

# Odborné prezentační nástroje Jupyter notebook slides a Mathematica slideshow notebook

Bc. Oleksandr Dolomanov

---

Bakalářská práce  
2021



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav informatiky a umělé inteligence

Akademický rok: 2020/2021

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Oleksandr Dolomanov**  
Osobní číslo: **A18687**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Softwarové inženýrství**  
Forma studia: **Prezenční**  
Téma práce: **Odborné prezentační nástroje Jupyter notebook slides a Mathematica slideshow notebook**  
Téma práce anglicky: **Professional Presentation Tools – Jupyter Notebook Slides and Mathematica Slideshow Notebook**

### Zásady pro vypracování

1. Vypracujte literární rešerši na dané téma.
2. Popište rozdíly mezi technologiemi Jupyter notebook slides a Mathematica slideshow notebook z praktického pohledu instalace, technických vlastností a využití pro distanční výuku.
3. Prověřte možnosti kooperace a interakce mezi vyučujícím a studenty.
4. Připravte ukázkové materiály ve formě výukových tutoriálů pro vybrané okruhy předmětů.
5. Vytvořte dokumentaci pro obsluhu a možnosti dalšího rozšiřování tutoriálů.



Forma zpracování bakalářské práce: **Tištěná/elektronická**

**Seznam doporučené literatury:**

1. Project Jupyter [online]. [cit. 2020-12-19]. Dostupné z: <https://jupyter.org/>
2. PECINOVSKÝ, Rudolf. Začínáme programovat v jazyku Python. Praha: Grada Publishing, 2020, 270 s. ISBN 978-80-271-1237-1.
3. PECINOVSKÝ, Rudolf. Python: kompletní příručka jazyka pro verzi 3.9. Praha: Grada Publishing, 2020, 478 s. Knižovna programátora. ISBN 9788027112692.
4. Wolfram Language [online]. [cit. 2020-12-19]. Dostupné z: <http://wolframlanguage.org/>
5. WOLFRAM, Stephen, [2017]. An elementary introduction to the Wolfram language. Second edition. Champaign: Wolfram Media. ISBN 1944183051.
6. TROTT, Michael. The Mathematica guidebook for symbolics. New York: Springer, c2006, xxxviii, 1453 s. ISBN 0387950206.
7. FRIEDRICH, Václav, 2013. Mathematica na počítači pro nematematiky. Ostrava: VŠB-TU Ostrava. ISBN 978-80-248-3162-6.

Vedoucí bakalářské práce: **doc. Ing. Roman Šenkeřík, Ph.D.**  
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **15. ledna 2021**

Termín odevzdání bakalářské práce: **17. května 2021**

**doc. Mgr. Milan Adámek, Ph.D. v.r.**  
děkan



**prof. Mgr. Roman Jašek, Ph.D. v.r.**  
ředitel ústavu

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 16.5.2021

Oleksandr Dolomanov v. r.  
.....  
podpis studenta

## **ABSTRAKT**

Hlavním cílem této bakalářské práce je představení technologií Jupyter notebook slides a Mathematica slideshow notebook a jejich srovnání z praktického pohledu instalace, technických vlastností a využití pro prezenční a distanční výuku. Cílem teoretické části této práce je poskytnout přehled v technologiích Jupyter notebook slides, Mathematica slideshow notebook a jazyka pro programování Python a Wolfram Language pro začátečníky. V praktické části jsou uvedeny instrukce pro instalaci Jupyter notebook a Wolfram Mathematica, ukázka způsobu vytvoření dokumentu prezentace v Jupyter notebook a Wolfram Mathematica a detailně představen postup vytvoření prezentace pomocí každé z technologií.

Klíčová slova: prezentace, Jupyter, Mathematica, notebook, JupyterHub, e-learning.

## **ABSTRACT**

The main goal of this bachelor thesis is to introduce the Jupyter notebook slides and Mathematica slideshow notebook and to compare them from the point of view of installation, technical properties and usability for distance learning. The aim of the theoretical part of this work is to provide an overview of the technologies Jupyter notebook slides and Mathematica slideshow notebook and the programming languages Python and Wolfram Language for all beginners. The practical part contains instructions for installing Jupyter notebook and Wolfram Mathematica, an example of how to create a presentation document in each one and a detailed procedure for creating a presentation using each technology.

Keywords: presentation, Jupyter, Mathematica, notebook, JupyterHub, cell, studying.

Děkuji svému vedoucímu bakalářské práce panu doc. Ing. Romanu Šenkeříkovi Ph.D za odborné vedení, za jeho připomínky a náměty. Také děkuji všem spolužákům, kteří mi svou kritikou pomohli ke zkvalitnění této práce.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

ÚVOD .....	8
I.....	9
TEORETICKÁ ČÁST .....	9
<b>1 VZDĚLÁVÁNÍ V DIGITÁLNÍ DOBĚ.....</b>	<b>10</b>
<b>2 TECHNOLOGIE MATHEMATICA SLIDESHOW NOTEBOOK.....</b>	<b>11</b>
2.1 PROGRAMOVACÍ JAZYK WOLFRAM: HISTORIE, POUŽÍTI, POPULARITA.....	11
2.2 ÚVOD DO WOLFRAM MATHEMATICA .....	12
2.2.1 Poznámkové bloky a buňky .....	12
2.2.2 Přesnost.....	12
2.2.3 Zadávání vzorců.....	12
2.2.4 Jednořádkové programování.....	13
2.2.5 Dynamické interaktivní výpočty .....	13
<b>3 TECHNOLOGIE JUPYTER NOTEBOOK.....</b>	<b>14</b>
3.1 PROČ JSOU NOTEBOOKY JUPYTER DOBRÉ PRO UČENÍ .....	14
3.1.1 Jednoduchost pro začátečníky.....	14
3.1.2 Rychlejší učení .....	15
3.1.3 Pomocný obsah spolu s kódem .....	15
3.1.4 Budoucí využitelnost prostředí .....	15
3.2 PROSTŘEDÍ PRO PRÁCI S JUPYTER NOTEBOOK .....	15
3.2.1 Jupyter na vzdáleném serveru.....	16
3.2.2 Spuštění v dočasném prostředí v cloudu .....	17
3.2.3 Vlastní serverová infrastruktura – JupyterHub .....	18
3.2.4 Spuštění Jupyter v cloudu.....	19
3.3 VOILÁ PRO NOTEBOOKY JUPYTER.....	21
3.4 JAZYK PYTHON.....	21
3.4.1 Použití programovacího jazyka Python.....	21
3.4.2 Historie jazyka python.....	21
3.4.3 Výhody Pythonu.....	22
3.4.4 Nevýhody Pythonu.....	22
3.4.5 Využití Pythonu.....	22
II.....	24
PRAKTICKÁ ČÁST .....	24
<b>4 ÚVOD DO PRAKTICKÉ ČÁSTI.....</b>	<b>25</b>

<b>5</b>	<b>WOLFRAM MATHEMATICA .....</b>	<b>26</b>
5.1	INSTALACE WOLFRAM MATHEMATICA .....	26
5.2	VYTVOŘENÍ PREZentaČNÍHO DOKUMENTU VE WOLFRAM MATHEMATICA .....	26
5.2.1	<i>Příklad vytvoření prezentace pro obor Matematická informatika.....</i>	<i>30</i>
<b>6</b>	<b>JUPYTER NOTEBOOK.....</b>	<b>38</b>
6.1	INSTALACE JUPYTER NOTEBOOK.....	38
6.2	KONFIGURACE VLASTNÍHO VÍCEUŽIVATELSKÉHO SERVERU JUPYTERHUB (V DOCKER).....	38
6.3	JAK VYTVOŘIT PREZentaČNÍ DOKUMENT V JUPYTER NOTEBOOKU .....	40
6.3.1	<i>Interaktivita v Jupyter.....</i>	<i>41</i>
6.3.2	<i>Příklad vytvoření prezentačního dokumentu v Jupyteru pro předmět matematika .....</i>	<i>42</i>
	<b>ZÁVĚR.....</b>	<b>48</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>50</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>52</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>53</b>



## ÚVOD

V případě nutnosti tvorby prezentace, člověka automaticky napadne několik typických prezentačních programů, jakými jsou například: Microsoft PowerPoint, Google Slides, Prezi. MPP a Google Slides. Tyto aplikace jsou samozřejmě velmi praktické pro vytváření prezentace. Nabízejí různé možnosti formátování, stylování a dobře se integrují s dalším softwarem vyvinutým společnostmi Microsoft a Google.

Existuje však několik dalších prezentačních programů, které jsou obzvlášť zajímavé pro ty, jenž potřebují předvést, jak funguje kód nebo graf matematické funkce. Těmi nástroji jsou Jupyter Notebook Slides a Mathematica slideshow. Většina programátorů zná notebooky Jupyter a Mathematica jako prostředí pro kódování, ale s nimi toho dokážou ještě více.

Práce je rozdělena do dvou částí. Teoretická část je zaměřena na charakteristiku technologií Jupyter Notebook Slides a Mathematica slideshow a popisuje, jaké programovací jazyky se v nich používají.

Praktická část popisuje problematiku instalace, samotnou tvorbu a editaci notebooku a možnosti jejich kooperace.

Účelem práce je provést analýzu a srovnání dvou programů pro vytváření a zobrazování prezentací s výpočetním jádrem notebooků Jupyter a notebooků Mathematica slideshow, a to z praktického hlediska instalace, technických vlastností a použití pro prezenční a distanční výuku.

V práci je prověřeno, zda dané programy podporují kooperaci, popřípadě jakým způsobem lze toho dosáhnout. Práce také ukazuje proces vytváření prezentací v těchto programech pro předmět Matematická informatika studovaný na UTB, aby jasně demonstroval všechny funkce, které mohou být užitečné ve vzdělávacím procesu, zejména pokud mohou zlepšit spolupráci a interakci mezi učitelem a studenty během distanční výuky. Vypracované tutoriály mohou sloužit pro další rozšíření nebo vytvoření nových prezentací pomocí těchto technologií na libovolné téma.

## **I. TEORETICKÁ ČÁST**

## 1 VZDĚLÁVÁNÍ V DIGITÁLNÍ DOBĚ

Rychlé tempo moderního života se odráží ve vzdělávání. Potřeba sdělit studentům více znalostí nevyhnutelně snižuje čas strávený nad základními kurzy. Za těchto podmínek je nutné zavést nejnovější technologie do výuky složitých, zejména technických disciplín.

S rozvojem informačních technologií se objevily elektronické vzdělávací zdroje. Nejprve se jednalo o elektronické kopie poznámek z přednášek přeložených do formátů webových stránek (html), dokumentů Microsoft Word (doc) nebo Adobe Acrobat (pdf). Vývoj prezentačního vybavení a zvýšení jeho dostupnosti, především z hlediska ceny, dalo podnět k vývoji prezentací ve formátu Microsoft PowerPoint (.ppt). Podle názoru autorů jsou však tato řešení nižší než klasické schéma přednášení křídou na tabuli. „Dynamika“ výukových materiálů mizí. Velké množství statických informací na obrazovce během přednášky otupuje pozornost studenta, dává mu pocit, že vše lze číst také v knize, a v důsledku toho vede k odmítnutí materiálu a úplné ztrátě zájmu o kurz [1].

Vypracování dynamických prezentací vyžaduje obrovské množství času na organizaci, někdy velmi složité propojení objektů. To zpravidla vede k nepořádku pomocných prvků na snímku, což znesnadňuje čtení v režimu úprav. Nejdůležitější je, že celá „dynamika“ se redukuje na dávkovanou prezentaci informací, což samozřejmě umožňuje zlepšit vnímání informací studenty, ale nedává kvalitativní podnět k porozumění uvažovaným předmětům [2].

Dalším problémem je kontrola znalostí studentů. Vypracování velkého počtu testů může být časově náročné. Současně lze vzhledem k dobrému technickému vybavení dnešních studentů tyto úkoly kompromitovat poměrně rychle, což bude vyžadovat jejich aktualizaci.

Také v případě velkých toků může být kontrola řešení problémů věnována dlouhou dobu, což znamená, že je racionálnější automatizovat tento proces [3].

Moderní úroveň softwaru samozřejmě umožňuje učitelům s dobrým počítačovým tréninkem řešit zadané úkoly pomocí specializovaného softwaru. Výsledkem však je sada různých softwarů, které někdy fungují v nekompatibilních formátech. Přednášky v jednom formátu, prezentace v jiném, problémy ve třetím, interaktivní test ve čtvrtém. To vše nutí učitele trávit čas převodem formátů, přenosem informací z jednoho programu do druhého atd.[4]

## 2 TECHNOLOGIE MATHEMATICA SLIDESHOW NOTEBOOK

Wolfram Mathematica je software, nejen pro matematické výpočty, ale obsahuje mnohem více možností, od modelování a simulace, vizualizace, dokumentace až po vytváření webových stránek. Software Mathematica má dále schopnost provádět volání funkcí a přijímat hovory z jazyků C, .NET, Java a dalších jazyků, generovat kód C, kompilovat samostatné knihovny a spustitelné soubory.

O všech výhodách software Mathematica si lze přečíst na oficiálních stránkách.<sup>1</sup>

Aby bylo možné používat software Mathematica, je nutné si jej stáhnout a nainstalovat do svého počítače. Mathematica funguje na Windows, Mac, Linux [5].

Wolfram Mathematica může být rovněž univerzálním balíčkem pro učitele pro vytváření přednášek, článků, modulů samostudia, testovacích problémů a interaktivních testů. Kromě provádění symbolických a numerických výpočtů Mathematica umožňuje vytvářet interaktivní články a prezentace [1]. Toho je dosaženo rozsáhlými schopnostmi systému pro vizualizaci různých dat: grafy, histogramy, matice a pole, obrázky, videa a grafické animace [2]. Vytvořené dokumenty lze exportovat v různých formátech. Seznam formátů souborů podporovaných programem Mathematica má 168 možností pro import a 138 pro export [3]. Výsledkem je, že se systém Wolfram Mathematica může stát hlavním nástrojem pro celý vzdělávací proces: od přípravy přednášek až po provádění zkoušek [4].

### 2.1 Programovací jazyk Wolfram: historie, použití, popularita

Wolfram je programovací jazyk, vyvinutý společností Wolfram Research pro systém Mathematica. Tento jazyk byl navržen jako nejuniverzálnější jazyk s důrazem na symbolické výpočty, funkčnost a logické programování s podporou libovolných datových struktur.

Do jazyka je zabudováno poměrně velké množství funkcí z různých oborů, například jsou zde zabudované funkce pro vytváření a provozování Turingova stroje, vytváření grafiky a zvuku, analýzu trojrozměrných modelů a řešení diferenciálních rovnic.

---

<sup>1</sup> <https://www.wolfram.com>

Jazyková dokumentace je rozsáhlá a jazyk není standardizovaný; je plánována částečná standardizace<sup>2</sup>.

Wolfram byl představen roku 1988, přesněji dne 28. květen, kdy dané datum je považováno za narozeniny daného jazyka. Do června 2013 jazyk neměl svůj oficiální název, místo toho se interně ve Wolframu používaly názvy „M“ a „jazyk Wolfram“ a uživatelé po systému, pro který byl vyvinut, často tento jazyk označovali jako „Mathematica“. Byly zvažovány varianty názvů „Lingua“ a „Express“, ale nakonec byl vybrán název, který se shoduje s názvem společnosti [6].

## 2.2 Úvod do Wolfram Mathematica

### 2.2.1 Poznámkové bloky a buňky

Všechny výpočty v Mathematice<sup>3</sup> jsou v poznámkových blocích a mají příponu nb. Poznámkové bloky obsahují kód i výsledky výpočtů a jsou rozděleny do různých typů buněk:

- Vstupní buňky – určují příkazy, které se budou počítat
- Buňky výsledků – zobrazují výsledek výpočtů
- Další buňky – buňky s textem, hlavičkami a vším ostatním

### 2.2.2 Přesnost

Mathematica dává přesnou odpověď, kdykoli je to možné. Pomocí funkce `N` je možné zobrazit numerickou aproximaci přesného výsledku. V Mathematice „přesnost“ ve skutečnosti znamená počet důležitých číslic. Přesnost výsledku můžete určit pomocí druhého argumentu funkce `N`.

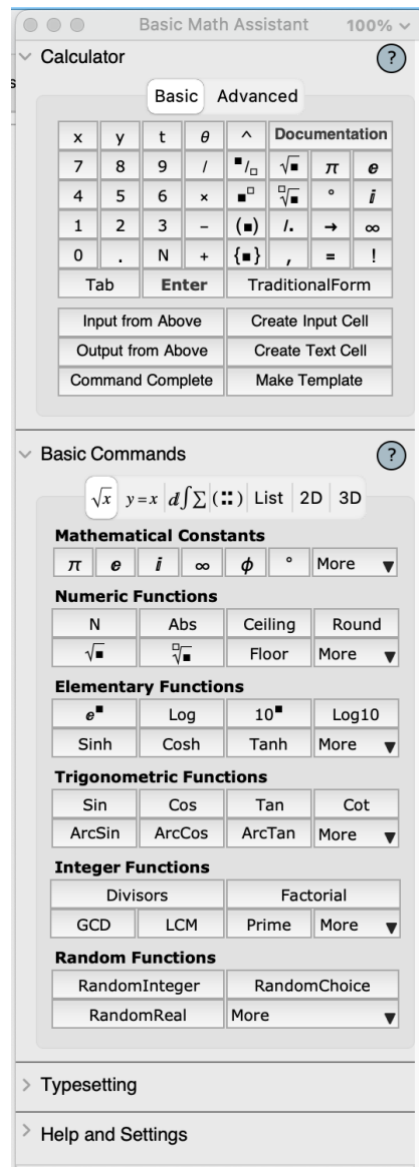
### 2.2.3 Zadávání vzorců

Mathematica poskytuje pohodlné zadávání vzorců. V rámci úvodního seznámení s funkcionalitou a zápisem v mathematica, mohou být palety výhodné pro používání (viz. Obrázek 1). Každé tlačítko na paletě má své vlastní klávesové zkratky.

---

<sup>2</sup> <https://reference.wolfram.com/language/>

<sup>3</sup> <https://www.wolfram.com/language/elementary-introduction/2nd-ed/>



Obrázek 1. Základní matematický asistent

### 2.2.4 Jednořádkové programování

Mathematica má mnoho běžných programovacích funkcí, například For, If, Switch. Není však je potřeba používat, pokud to není nezbytně nutné, vzhledem k tomu, že téměř vše stejné lze provést v jednom řádku pomocí speciálních funkcí a jejich kombinací (zpočátku může být obtížné reorganizovat takový programovací styl).

### 2.2.5 Dynamické interaktivní výpočty

Jednou z výhod vlastností Mathematica je dynamická práce na počítači. Umožní manipulovat s daty a sledovat, jak se výsledek dynamicky mění.

Pro dynamické výpočty se používají funkce Dynamic, Manipulate atd. [6]

### 3 TECHNOLOGIE JUPYTER NOTEBOOK

Jupyter Notebook je interaktivní webová aplikace s otevřeným zdrojovým kódem, která umožňuje psát a spouštět kód ve více než 40 programovacích jazycích, včetně Pythonu, R, Julie a Scaly. Notebook Jupyter je produkt Project Jupyter, který je velmi užitečný pro iterativní programování, neboť umožňuje napsat malý kousek kódu, spustit jej a odeslat výsledek.

Notebook Jupyter umožňuje vytvářet dokumenty ve tvaru notebooku. Notebooky Jupyter jsou publikované a reprodukováné výzkumné dokumenty obsahující prvky RTF, rovnice, kód a výsledky provádění (obrázky, tabulky, interaktivní grafy). Notebooky lze exportovat do dokumentů HTML nebo PDF a lze je použít k vytváření interaktivních prezentací nebo webových stránek.

Notebooky Mathematica a Jupyter jsou vizuálně velmi podobné, i když jsou navrženy pro práci s různými programovacími jazyky a pro práci v různých vývojových prostředích. Wolfram Mathematica, stejně jako Jupyter Notebook, nabízí možnost zobrazit kód jako prezentace. Lze k nim přidávat nadpisy a komentáře ve formě textových bloků, které jsou ideální pro rozdělení kódu do bloků a popis těchto bloků kódu.

#### 3.1 Proč jsou notebooky Jupyter dobré pro učení

Notebooky Jupyter jsou skvělým nástrojem nejen pro vývojáře a datové vědce, ale také pro výuku.

Notebooky Jupyter jsou ideální pro učení, neboť:

1. Eliminují složitost pro začátečníky.
2. Umožňují rychlejší interakci a tím i rychlejší učení.
3. Umožní vložit pomocný obsah vedle buněk kódu.
4. Umožňují studentům učit se ve stejném prostředí, které později použijí.

##### 3.1.1 Jednoduchost pro začátečníky

Notebook Jupyter v zásadě odstraňuje veškerou složitost začátků s jakýmkoli typem kódování na vysoké úrovni. Tento způsob je obzvláště výhodný a praktický pro začátečníky, kdy je jim umožněno jednoduše přejít do Jupyteru, vytvořit notebook, začít psát a spouštět kód. Je to jistě lepší způsob než způsob tradiční, kdy by si studenti museli dělat starosti s

podrobnostmi používání terminálu, nastavením prostředí, instalací knihoven, logiku a syntaxi a umožňuje učitelům soustředit se více na výuku.

### 3.1.2 Rychlejší učení

Jednou ze zásadních věcí rozhraní notebooku je, že umožňuje uživateli psát a spouštět kód v libovolném pořadí a rychlým tempem. Jen malý počet lidí dokáže napsat velké množství kódů zcela bez chyb, proto se obvykle napíše malý úsek kódu a otestuje se. Poté jsou postupně přidávány nové části s testováním v každé iteraci. Rozhraní notebooku je pro tento proces praktické, neboť poskytuje okamžité testování, aniž byste museli znovu spouštět celý skript. Ukázalo se, že je to rovněž příznivé pro studium, protože umožňuje studentům získat rychlou zpětnou vazbu, čímž se dynamicky zvyšuje jejich rychlost učení.

### 3.1.3 Pomocný obsah spolu s kódem

Jelikož buňky v poznámkových blocích lze použít k zápisu Markdown nebo k zápisu kódu, který zobrazuje obsah LaTeX nebo HTML, mohou instruktoři připravit notebooky, které hostují pomocný obsah, jako je jednoduchý text s vysvětlením, odkazy na externí zdroje, matematické rovnice, obrázky, dokonce i video vedle buněk, kde musí student napsat a spustit kód. To může pomoci zjednodušit proces poskytování pokynů v situacích, kdy instruktor není fyzicky přítomen se studentem, aby mu pomohl.

### 3.1.4 Budoucí využitelnost prostředí

Potenciální nevýhodou kódovacích prostředí pro vzdělávání je, že jsou určeny pouze pro výuku, kdy ve skutečnosti je nelze použít pro práci. Ve výsledku studenti tráví mnoho hodin zkoumáním prostředí, které poté nikdy nevyužijí. Vzhledem k tomu, jak rychle si Jupyter v oboru získává popularitu, se studenti mohou těšit na používání notebooků ve chvíli, kdy potřebují uplatnit dovednosti, které se ve společnosti naučili [7].

## 3.2 Prostředí pro práci s Jupyter Notebook

Existuje několik možností, jak distribuovat notebooky Jupyter studentům. Pro použití Jupyteru jej lze nainstalovat na osobní počítače, univerzitní počítače nebo na vzdálený internetový server, ke kterému studenti mají přístup.



„Lokální instalace“ znamená, že na každém počítači je spuštěn software, který obsahuje notebook Jupyter. To obvykle vyžaduje instalaci distribuce, jež zahrnuje Jupyter, Python a další jádra.

Populární softwarová distribuce, jež zahrnuje Jupyter, se nazývá Anaconda. Tu lze snadno nainstalovat na operační systémy Windows, Mac a Linux. Jelikož distribuce může instalovat vše s oprávněními na úrovni uživatele, nevyžaduje, aby měl uživatel k počítači přístup správce (nebo root). Anaconda zahrnuje více než 1 500 softwarových balíčků poskytujících většinu, ne-li veškerý, software potřebný pro studenty. Notebooky Jupyter lze otevřít spuštěním notebooku Jupyter nebo jejich otevřením prostřednictvím IDE Spyder. Díky těmto atributům je atraktivní jak pro osobní použití, tak pro instalaci do počítačů řízených institucemi.

### 3.2.1 Jupyter na vzdáleném serveru

Jupyter pracuje lokálně jako webová aplikace, což znamená, že běží v prohlížeči, který je připojený k serveru. V místní instalaci funguje prohlížeč a server na stejném počítači. Samotný server lze spustit také vzdáleně. V takovém případě není potřeba nic instalovat; stačí spustit prohlížeč a zadat adresu URL.

Existuje několik způsobů, jak spustit Jupyter na vzdáleném serveru:

1. Jupyter lze spustit na serveru, který vlastní jednatel či instituce.
2. Jupyter je možné spustit v dočasném prostředí spuštěném v cloudu.
3. Jupyter lze spustit v trvalém prostředí probíhající v cloudu.

Vzdálený průběh Jupyteru má oproti laboratorní práci mnoho výhod. Je možné poskytnout konzistentní prostředí a zajistit, aby studenti měli přístup k dostatečným výpočetním prostředkům. Dále odstraňuje jednu z hlavních nevýhod laboratorního nastavení, kdy studenti mají přístup ke cloudovým zdrojům odkudkoli, nejen na akademické půdě.

Práce v cloudu také znamená, že studenti již nemusí spravovat své vlastní zálohy na pevný disk notebooku. Ačkoliv student může omylem přepsat, odstranit nebo zničit obsah notebooku uloženého v cloudu, neztratí veškerou svou práci, pokud je notebook poškozen nebo ztracen. Pro jednoduché jednorázové použití Jupyteru je možnost cloudu velmi výhodná, protože instalace dalšího softwaru zabere málo času.

### 3.2.2 Spuštění v dočasném prostředí v cloudu

Nejjednodušší způsob, jak spustit Jupyter v cloudu, je použití cloudové služby, která poskytuje dočasná prostředí. Některé z těchto služeb jsou zdarma a mohou se používat, aniž by byla potřeba cokoli instalovat.

Tyto služby jsou dobré pro krátké příklady v učebnách, které Jupyter příliš nepoužívají. Studenti mohou notebook otevřít a spustit stisknutím jediného tlačítka.

Tyto služby však mají určitá omezení:

- Pokud notebooky závisí na konkrétních balíčcích nebo konkrétních verzích balíčků, může být splnění těchto požadavků obtížné.
- Služby spouštějí notebooky v dočasném prostředí, které zmizí, když se nepoužívají.
- Některé z těchto služeb nezaručují úroveň služeb a nemusí být tak spolehlivé.

#### 3.2.2.1 Binder

Binder je služba s otevřeným zdrojovým kódem, kterou poskytuje Project Jupyter. Umožňuje vlastníkově kolekce poznámkových bloků ve veřejném úložišti provést předběžný obrázek ve službě Binder a získat sdílený odkaz, který může každý návštěvník použít k získání fungující instance JupyterHub s předinstalovanými notebooky do úložiště. Relace je dočasná, všechny změny provedené uživatelem budou odstraněny po zavření záložky nebo okna, ale je plně interaktivní. Binder je v současné době jednou z oblíbených služeb pro jednorázové semináře nebo vzdělávací programy.

Proces nasazení notebooku na serveru Binder vyžaduje umístění notebooku ve veřejném úložišti GitHub, kdy samotný proces vypadá takto:

1. Vytvoření notebooku Jupyter
2. Přidání metadat pro zobrazení v prezentačním formátu (označení Cell jako Slide apod.)
3. Pro správnou funkci notebooku je nutné vytvořit soubor `environment.yml`, jehož obsahem budou všechny závislosti
4. Publikování souboru notebooku a `environment.yml` do GitHub
5. Na stránce <https://mybinder.org> třeba poskytnout odkaz na úložiště GitHub
6. Je doporučeno provést kontrolu na robota a zadat větev, značku nebo potvrzení

7. Klepnout na „Launch“
8. Výsledkem bude přesměrování na stránku s otevřeným notebookem Jupyter na serveru Binder se soubory z úložiště GitHub, které bylo uvedeno jako zdrojové, kde je stejné prostředí, jako při lokální instalaci. Je možnost si notebook spustit a otevřít živou prezentaci díky odkazu na RISE knihovnu v environment.yml

Soubor environment.yml může vypadat následovně:

```
name: example-environment
channels:
  - conda-forge
dependencies:
  - numpy
  - psutil
  - toolz
  - matplotlib
  - dill
  - pandas
  - partd
  - bokeh
  - dask
```

### 3.2.3 Vlastní serverová infrastruktura – JupyterHub

Pokud je k dispozici server nebo klastr s dostatečným výkonem pro zpracování všech požadavků třídy, včetně CPU a zejména paměti, je možné Jupyter poskytnout jako službu pomocí JupyterHub.

JupyterHub je software s otevřeným zdrojovým kódem, který poskytuje cloudovou aplikaci Jupyter pro každého uživatele ve skupině. Každý uživatel má na serveru svůj vlastní účet a domovský adresář. Hub, centrální systém JupyterHub, umožňuje uživatelům ověřovat a spouštět samostatné servery notebooků Jupyter. Programy, které provozují servery pro notebooky, mohou používat řadu technických řešení<sup>4</sup>.

Poté, co Hub spustí notebookový server uživatele, chová se notebook Jupyter spuštěný v cloudu stejně jako Jupyter nainstalovaný v individuálním počítači, ale JupyterHub spustí notebooky a uloží soubory na vzdáleném cloudovém počítači. Studenti si mohou stáhnout

---

<sup>4</sup> <https://github.com/jupyterhub/jupyterhub/wiki/Spawners>

notebooky uložené v cloudu do svého místního počítače, pokud chtějí pracovat také s místní instalací. Kromě toho mohou studenti nahrávat notebooky (a další soubory) ze svého místního počítače do cloudu.

I když může server JupyterHub provozovat kdokoli na svém vlastním počítači se systémem Linux nebo Mac, instalace a konfigurace JupyterHub vyžaduje komplexní znalosti zahrnující operační systém Linux / Unix, správu systému a síťové připojení.

Další informace viz:

- <https://github.com/jupyterhub/jupyterhub> (základní projekt JupyterHub, který lze nainstalovat na holý server, virtuální privátní server (VPS) nebo komerční cloudový cluster)
- <https://github.com/jupyterhub/the-littlest-jupyterhub> (zjednodušená instalace JupyterHub na vzdáleném serveru nebo VPS)
- <https://github.com/jupyterhub/zero-to-jupyterhub-k8s> (podrobný průvodce instalací JupyterHub do cloudového systému Kubernetes)

Poskytování služby JupyterHub má několik výhod. Zaprvé, studenti mohou ihned začít pracovat a neztrácejí čas instalací softwaru. Přechodem na URL se přihlásí do JupyterHub a začnou používat Jupyter. Tato schopnost rychle se přihlásit a začít používat počítač je účinným způsobem, jak zapojit studenty do lekce a vyhnout se někdy stresující zkušenosti s instalací softwaru na počítač.

Spuštění JupyterHub na vlastním serveru má však nevýhody. Pro začátečníky není spuštění snadné a většina školitelů bude potřebovat nebo alespoň těžit z institucionální podpory, která nemusí být k dispozici. Pokud se zvýší počet studentů, bude potřeba většího výpočetního výkonu. Pracovní zátěž, kterou studenti kladou, může být nerovnoměrná; například pokud všichni současně používají výpočetně intenzivní příklad, server to nemusí zvládnout.

Tato možnost může být nákladná, zejména pokud servery nemají dostatečnou kapacitu.

### 3.2.4 Spuštění Jupyter v cloudu

Pokud výpočetní hardware, schopný podporovat práci třídy, není k dispozici, je možné spustit JupyterHub na virtuálních serverech poskytovaných cloudovými službami, jako jsou AWS a Microsoft Azure. V tomto prostředí je možné nainstalovat JupyterHub.

Existují také komerční nabídky pro používání Jupyteru v cloudu, z nichž některé poskytují bezplatné zkušební verze nebo cenový model „freemium“.

Příklady:

- CoCalc (dříve SageMathCloud) (<https://cocalc.com>) je online výpočetní prostředí s otevřeným zdrojovým kódem s prémiovou podporou notebooků Jupyter udržovanou společností SageMath, Inc. Jedná se o jednu z mála služeb, která umožňuje více uživatelům upravovat notebook Jupyter současně. Lze jej použít zdarma a velké výpočetní zdroje je možné získat zaplacením měsíčního, ročního poplatku nebo poplatku za předplatné kurzu.
- HubHero (<https://hubhero.net>) poskytuje vyučujícím profesionálně nakonfigurované servery JupyterHub. Vlastníkem HubHero je Tim Head, vedoucí projektu na odkazu <https://mybinder.org>, který je přispěvatelem do JupyterHub. U kurzů s maximálním počtem 30 studentů je nabízeno komunitní nastavení, které poskytuje konfiguraci vlastního JupyterHub na hardwaru zákazníka nebo poskytovatele cloudu dle výběru. Pro větší kurzy nebo plně spravovaná nasazení jsou k dispozici také hostovaná řešení.

Největší výhodou těchto služeb je, že nevyžadují instalaci ani minimální konfiguraci instruktory. Některé z nich poskytují funkce, které se integrují do systémů pro správu učení. Vyučující však obvykle musí vytvářet studentské účty a nastavovat prostředí pro studenty.

Nevýhodou těchto služeb je, že mohou být drahé. Někteří poskytovatelé si účtují poplatek za studenta s výpočetním a paměťovým omezením. Někdy účtují poplatky na základě skutečného využití, což může být ale nepředvídatelné a hrozí, že učitelé omezí aktivity svých studentů. Mezi další nevýhody patří instalace požadovaných balíčků nebo konkrétních verzí balíčků, které však mohou být obtížné nebo nemožné. Některé z těchto služeb omezují to, co mohou studenti dělat; například mít omezenou schopnost přístupu k externím službám. Mnohé z těchto služeb jsou relativně nové a výhodné, jindy se naopak učitelé a studenti potýkají s obtížemi. Studenti obvykle ztratí přístup ke svým účtům na konci lekce nebo nějakou dobu po ní. Při výměně informací o studentech na komerčních serverech mohou nastat problémy s důvěrností. Některé instituce mají dohody o zachování důvěrnosti s jedním nebo více z těchto poskytovatelů [8].

### 3.3 Voilá pro notebooky Jupyter

Voilá je balíček Pythonu, který promění notebooky Jupyter na produkční weby.<sup>5</sup> Voilá podporuje interaktivní widgety Jupyter, včetně cest backhaul k jádru, ale neumožňuje spuštění libovolného kódu spotřebiteli řídicího panelu. Na základě standardních protokolů a formátů souborů Jupyter, pracuje Voilá s jakýmkoli jádrem Jupyter (C ++, Python, Julia), což z něj činí jazykově nezávislý systém ovládacího panelu. Voilá je rovněž rozšiřitelná a obsahuje flexibilní systém šablon pro vytváření různých rozložení aplikací [9]

### 3.4 Jazyk Python

#### 3.4.1 Použití programovacího jazyka Python

Python je aktivně se rozvíjející skriptovací jazyk, jenž se používá k řešení velkého množství nejrozličnějších problémů a úkolů. Python je užitečný při vytváření počítačových a mobilních aplikací, používá se při práci s velkým množstvím informací, při vývoji webových stránek a dalších různých projektů. Rovněž je využíván při strojovém učení. Tento programovací jazyk používají známé společnosti, jako jsou Spotify a Amazon (například k analýze dat a vytvoření doporučovacího algoritmu), YouTube, Instagram, a dokonce i Walt Disney. Python si tedy našel své místo v různých oblastech.

#### 3.4.2 Historie jazyka python

Vývoj jazyka Python začal na konci 80. let 20. století. Distribuovaný operační systém společnosti Amoeba vyžadoval rozšiřitelný skriptovací jazyk. Zaměstnanec nizozemského institutu Guido van Rossum vytvořil daný jazyk ve svém volném čase. Guido van Rossum vydal již v roce 1991 první kód. Vývojář pojmenoval jazyk po slavné britské humorné televizní show sedmdesátých let - „Létající cirkus Montyho Pythona“<sup>6</sup>.

---

<sup>5</sup> <https://github.com/voila-dashboards/voila>

<sup>6</sup> <https://www.csfd.cz/film/70220-monty-pythonuv-letajici-cirkus/prehled/>

### 3.4.3 Výhody Pythonu

Jazyk se vyznačuje logickou syntaxí, díky níž je zdrojový kód programů napsaných „v Pythonu“ snadno čitelný a srozumitelný. Další z výhod tohoto programovacího jazyka je jeho relativní snadnost. Považuje se za nejvhodnější pro začínající odborníky: po několika dnech studia se lze naučit vyvíjet jednoduché programy. Další výhodou je velká online komunita, což znamená, že pokud vývojář čelí otázkám a obtížím, může vždy požádat kolegy o radu, což výrazně urychluje řešení problémů. Výhodná je rovněž flexibilita a škálovatelnost. Python umožňuje vývojářům přizpůsobit aplikační logiku na vysoké úrovni, což usnadňuje rozšiřování složitých aplikací podle potřeby. Vývoj „v pythonu“ je rychlejší než ve většině ostatních jazyků. Python je interpretovaný programovací jazyk, což znamená, že se jedná o soubor ve formátu prostého textu před spuštěním, proto jej lze programovat téměř na všech platformách.

### 3.4.4 Nevýhody Pythonu

Nevýhodou je velká rychlost algoritmu. Může být těžké psát vysoce výkonné projekty v čistém Pythonu, což vyžaduje uchýlení se k pomoci jiných jazyků. Zabezpečení poskytované paměťovým modelem Pythonu neguje většinu možných optimalizací procesoru [10].

### 3.4.5 Využití Pythonu

Python se nejčastěji používá při vývoji webu a analýze velkých dat. K doplnění funkčnosti jazyka se používají různé rámce: Django, Pyramid, Flask a další. Python je vhodný také pro vytváření aplikací nebo her, například grafický editor GIMP je napsán v Pythonu. V tomto jazyce je vyvíjen také torrentový klient BitTorrent až do verze 6. Python byl také použit při vývoji herních projektů třídy AAA: EVE Online <sup>7</sup>, Battlefield 2 <sup>8</sup>, World of Tanks <sup>9</sup> a další.

Nejčastěji je v takových případech jedna ze součástí projektu napsána v Pythonu, tzn., že pouze nějaký druh modulu je psán v Pythonu, například na straně serveru. Jazyk se rovněž používá při správě systému k automatizaci úkolů. Využívá se téměř na všech linuxových

---

<sup>7</sup> <https://www.eveonline.com>

<sup>8</sup> <https://www.ea.com/games/battlefield/battlefield-2>

<sup>9</sup> <https://worldoftanks.eu>

serverech. „Python“ je velmi vhodný pro práci s daty ve vědeckém výzkumu. Algoritmy strojového učení a analýzy dat jsou psány v tomto jazyce [11].

Podle zprávy RedMonk z ledna 2020 se Python stal druhým nejpopulárnějším programovacím jazykem po Java Script. Dříve tuto pozici s jistotou dlouho držela Java, ale na začátku roku se tato Java posunula na třetí řádek hodnocení, který je tvořen na základě informací z repozitářů GitHub. Přesněji řečeno, dvojice Java Script a Java si udržely první oblíbenost dvou programovacích jazyků od začátku formování specifikovaného hodnocení, tedy od roku 2012 [12].



## **II. PRAKTICKÁ ČÁST**

## 4 ÚVOD DO PRAKTICKÉ ČÁSTI

Obsahem praktické části je instalace a konfigurace Jupyter Notebook a Wolfram Mathematica. Praktická část popisuje, jak v těchto programech vytvářet, případně editovat prezentace, a také příklady vytváření prezentací pro předmět Matematická informatika studovaný na UTB.

Praktická část má následující strukturu:

1. Wolfram Mathematica
  - Instalace Wolfram Mathematica
  - Jak vytvořit prezentační dokument ve Wolfram Mathematica
  - Příklad vytvoření prezentace pro Matematická informatika
2. Jupyter Notebook
  - Instalace Jupyter Notebook
  - Jak vytvořit prezentační dokument ve Jupyter Notebook
  - Interaktivitu
  - Příklad vytvoření prezentace pro Matematická informatika

## 5 WOLFRAM MATHEMATICA

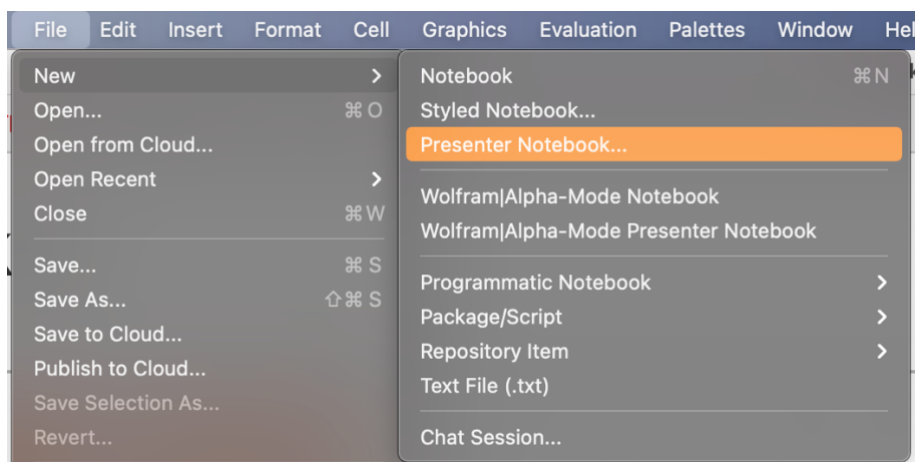
### 5.1 Instalace Wolfram Mathematica

Pro instalaci Wolfram Mathematica je potřeba se zaregistrovat na oficiálních webových stránkách společnosti<sup>10</sup>.

Dále je nutné koupit desktopovou verzi programu, ve svém osobním účtu na webu stáhnout program a provést klasickou instalaci, která se neliší od instalace jiných programů. Během instalace je třeba potvrdit přítomnost licence uvedením aktivačního klíče, který je k nalezení v osobním účtu na oficiálních webových stránkách společnosti.

### 5.2 Vytvoření prezentačního dokumentu ve Wolfram Mathematica

1. Pro vytvoření dokumentu prezentace z nabídky “File” dokument, zvolit New => Presenter Notebook (viz. Obrázek 2. Volba typu nového dokumentu)

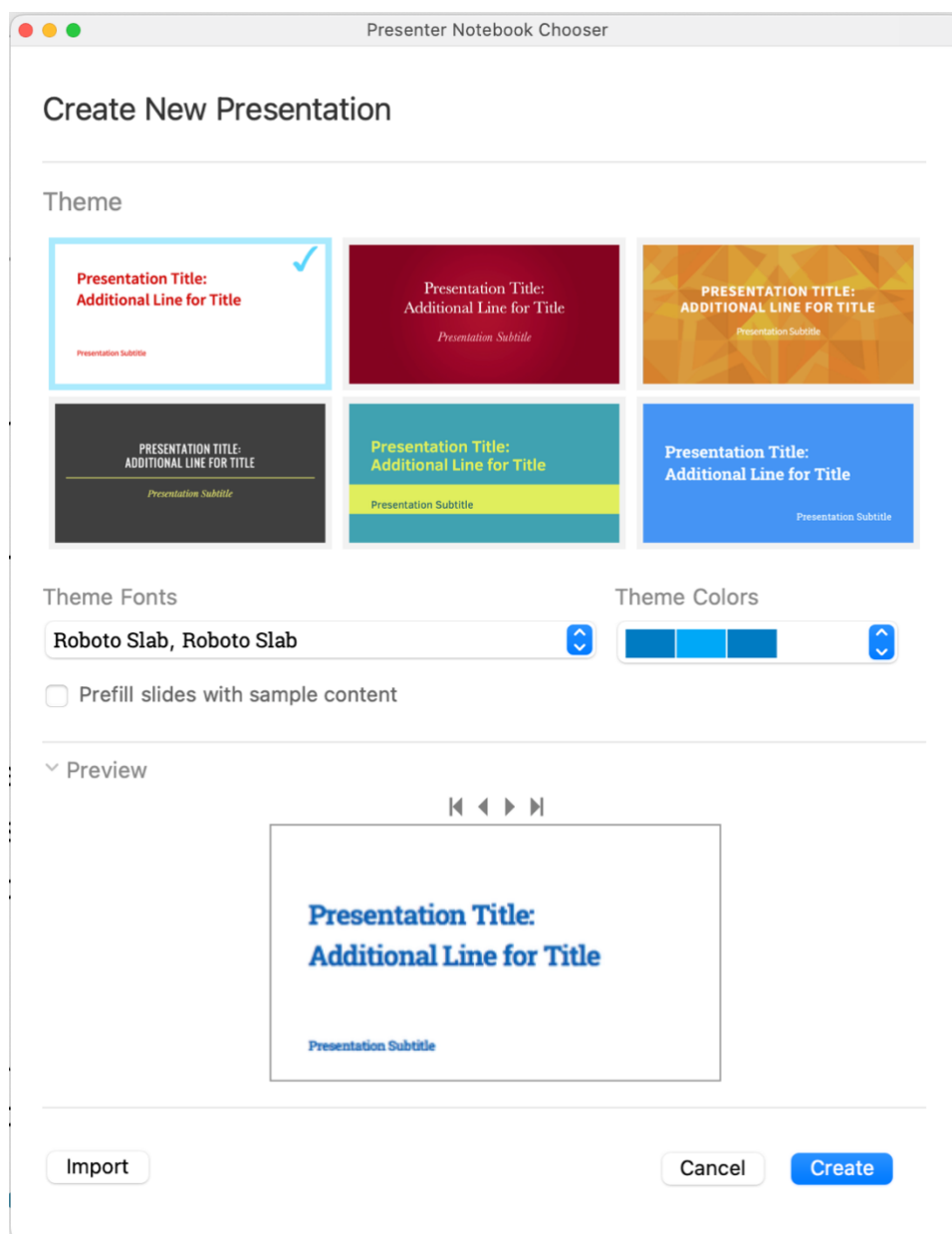


Obrázek 2. Volba typu nového dokumentu

2. Následuje možnost vybrat motiv, písmo a barvu. Po vybrání tématu prezentace a písma lze prezentaci vytvořit pomocí “Create” (Viz. Obrázek 3. Základní nastavení prezentačního dokumentu)

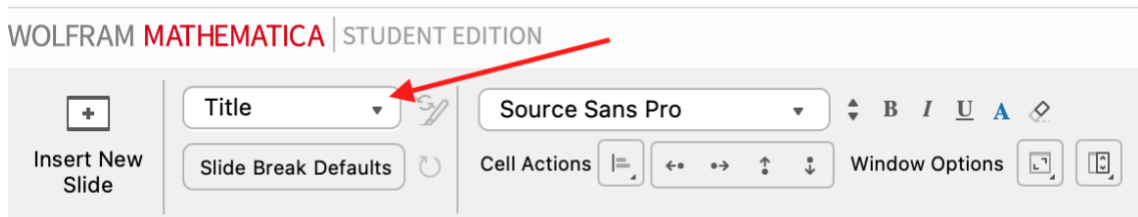
---

<sup>10</sup> <https://account.wolfram.com/login/create>

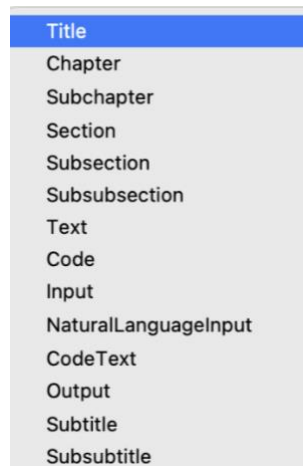


Obrázek 3. Základní nastavení prezentačního dokumentu

3. Pro vložení nového prvku do slidu (viz. Obrázek 4. Menu pro přidání nového prvku slidu), je třeba použít nabídku pro vložení prvků (viz. Obrázek 5)

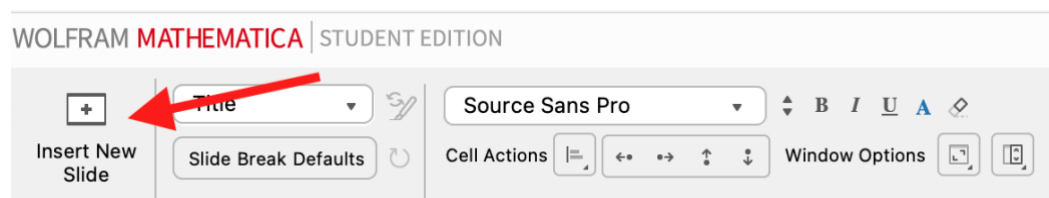


Obrázek 4. Menu pro přidání nového prvku slidu



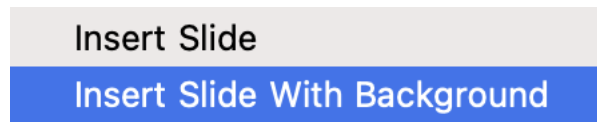
Obrázek 5. Nabídka prvku pro přidání do slidu

4. Zajímavou funkcí je možnost v Mathematica Slideshow Notebook přidat prvek pro zadávání a spouštění kódu v jiných programovacích jazycích než Wolfram. K tomu lze použít prvek „ExternalLanguage“. Podporováno: Python, Ruby, R, SQL, SQL – JDBC, Julia, NodeJS
5. Pro vložení nového slidu, je nutné použít tlačítko „Insert New Slide“ (Viz. Obrázek 6. Tlačítko pro přidání nového slidu).



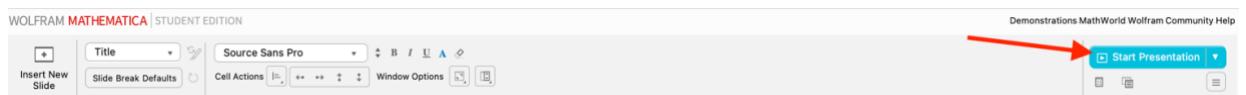
Obrázek 6. Tlačítko pro přidání nového slidu

Z rozbalovací nabídky je možné vybrat běžný slide nebo slide s pozadím (viz. Obrázek 7. Nabídka typu nového slidu)



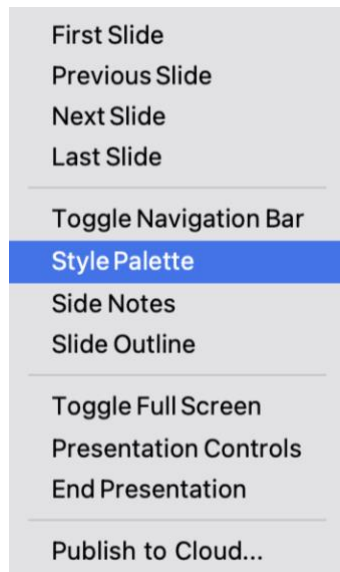
Obrázek 7. Nabídka typu nového slidu

6. Prezentaci lze otevřít kliknutím na „Start Presentation“ v pravém horním rohu (viz. Obrázek 8)



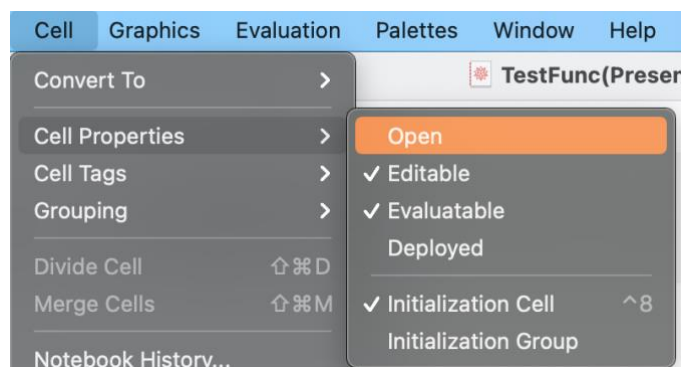
Obrázek 8. Tlačítko pro spuštění prezentace

7. Pro zavření prezentace je třeba rozkliknout menu v pravém horním rohu a vybrat „End Presentation“
8. Přepínání dozadu a dopředu se provádí pomocí šipek vlevo a vpravo
9. Pro změnu formátu zobrazení prezentace je možné rozkliknout vypadávající nabídku kliknutím na šipku na pravé straně tlačítka „Start Presentation“
10. Pro pohodlnější změny stylu slidu je možné použít „Paletu stylů“ přímo v režimu prezentace. Pro její otevření je potřeba kliknout na tlačítko menu v pravém horním rohu a vybrat z nabídky „Style Palette“ (Viz. Obrázek 9. Menu pro správu prezentace).



Obrázek 9. Menu pro správu prezentace

11. Často je nutné skrýt spustitelný kód a zobrazit pouze výsledek, například skrýt kód pro vykreslení a zobrazit pouze vykreslený graf. Toho lze dosáhnout výběrem buňky s funkčním kódem a v menu Cell => Cell Properties vypnout “Open” a zapnout “Initialization Cell” (Viz. Obrázek 10. Menu pro správu buňky).

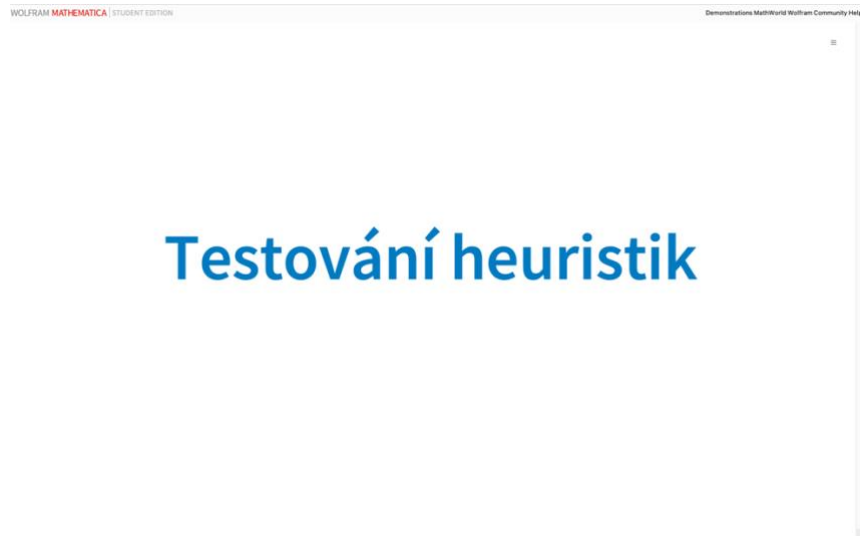


Obrázek 10. Menu pro správu buňky

### 5.2.1 Příklad vytvoření prezentace pro obor Matematická informatika

1. Nejdříve se vytvoří prezentační dokument.
2. Pro vložení slide, jen s názvem prezentace, je možné použít postup:
  - Přidáme element Title, jeho obsahem bude: “Testování heuristik”

Po zarovnání na střed by měl první slide vypadat tak, jak je uvedeno na Obrázek 11.



Obrázek 11. Záhlavní slide prezentace v Mathematica slideshow notebook

3. Druhý slide prezentace bude obsahovat seznam definicí. Pro vložení slidu s odrážkami použijeme postup:
  - Přidáme element Section pro nadpis, jeho obsahem bude: “Testování heuristických algoritmů”
  - Přidáme element Item, jeho obsahem bude: “Pro testování optimalizačních algoritmů se používají dva rozdílné postupy”
  - “Item”: “V prvním postupu obvykle vycházíme z již existujících příkladů, které již byly řešeny jinými algoritmy. Výsledky právě testovaného algoritmu pak porovnáme s výsledky již existujícími (publikovanými v literatuře)”
  - “Item”: “Druhý způsob spočívá v tom, že použijeme speciálně sestavenou množinu testovacích funkcí (tzv. benchmark set nebo také testbed), obsahující funkce s různými vlastnostmi, kombinace unimodálních, multimodálních, hybridních a kompozitních funkcí”
4. Další příklad odrážkového slidu:



- “Item”: “Vzhledem k tomu, že jsou známy analytické vztahy, je u většiny z nich velmi jednoduché vypočítat pozici a hodnotu extrému pro libovolnou dimenzi”.
- “Item”: “Pro profesionální testování se používají testovací sady, kupříkladu pro IEEE CEC Competitions nebo GECCO kongres”.

5. Příklad slidu s nadpisem a odrážkou:

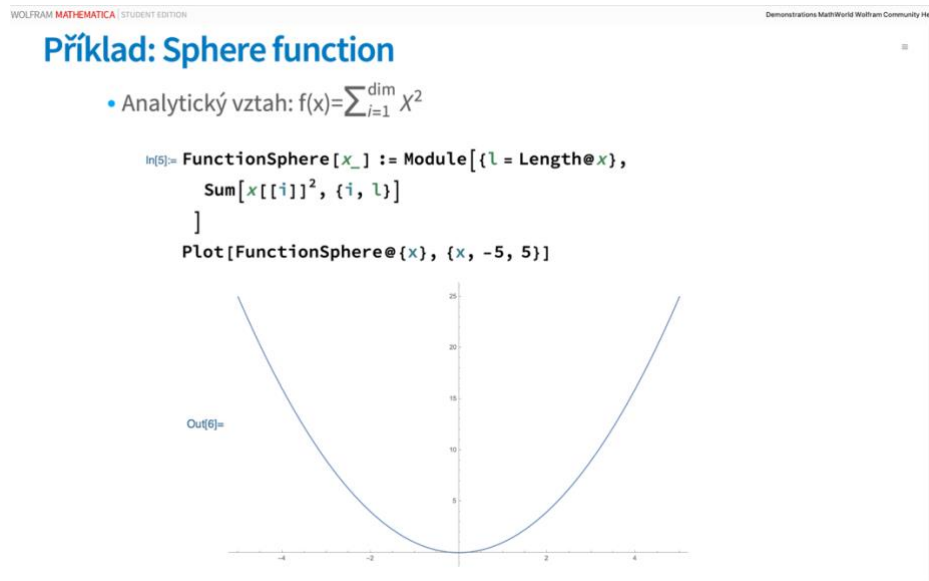
- “Section”: “Kreslení funkcí v Matematice”
- “Text”: “Pro vytváření 2D a 3D grafů má Mathematica zabudované funkce Plot a Plot3D. Dvourozměrný graf se skládá z nezávislé proměnné (x) a závislé proměnné (y), což je hodnota objektivní funkce. Trojrozměrný graf má obě nezávislé proměnné”

6. Na dalším slidu se ukazuje způsob definice „Sphere function“ a použití funkce Plot. Název sekce dalšího slidu bude: “Příklad: „Sphere function”:

- “Item”: “Analytický vztah:  $f(x) = \sum_{i=1}^{dim} x^2$ ,”
- “Input”:

```
FunctionSphere[x_] := Module[{l = Length@x},  
  Sum[x[[i]]^2, {i, l}]  
]  
Plot[FunctionSphere@{x}, {x, -5, 5}]
```

Na Obrázek 12 je vidět výsledný slid, který obsahuje text, input pole pro spustitelný kód a výsledný grafický výstup kódu.



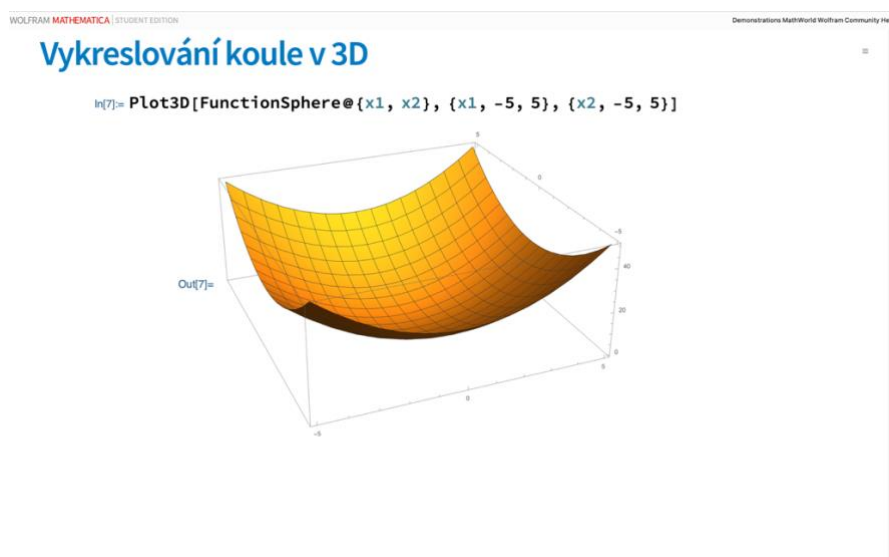
Obrázek 12. Příklad použití funkce Plot ve Wolfram

7. Na dalším slajdu je představeno použití funkce Plot3D.

- “Section”: “Vykreslování koule v 3D”
- “Input”:

```
Plot3D[FunctionSphere@{x1, x2}, {x1, -5, 5}, {x2, -5, 5}]
```

(Viz. Obrázek 13)



Obrázek 13. Příklad použití funkce Plot3D

8. Obsahem dalšího slidu je definice Rastriginove funkce a její vykreslení pomocí funkci Plot:

- “Section”: “Rastriginova funkce”

- “Item”: “Analytický vztah:  $2D\sum_{i=1}^D x_i^2 - 10\cos(2\pi x_i)$ ”

Pro vytvoření vzorců lze použít „Basic Math Assistant“, která se nachází v menu „Palettes“.

- “Input”:

```
RastriginFunction[x_] := Module[{d},  
d = Length@x;  
10 d + Total[#^2 - 10 Cos[2 \[Pi] #] & /@ x];  
Plot[RastriginFunction@{x}, {x, -5, 5}]
```

9. Na dalším slidu si definujeme funkce pro další použití v Manipulate. Tento slajd může být skryt.

- “Input”:

```

SphereFunctionPrint[d_] :=
Module[{min, max, function, minValue, params, interval, step},
  min = -5.12; max = 5.12; step = 0.1;
  function =
    "\!\(\*\UnderoverscriptBox[\(\[Sum]\), \(\i = 1\), \
\(\d\)]\)\!\(\*\SuperscriptBox[\(\x\), \(\2\)]\)\)";
  interval =
    "x \[Epsilon] <" <> ToString@min <> " ; " <> ToString@max <>
">";
  minValue = 0; params = 0;
  {min, max, interval, function, minValue, params, FunctionSphere,
  step}
];
RastriginFunctionPrint[d_] :=
Module[{min, max, function, minValue, params, interval, step},
  min = -5.12; max = 5.12; step = 0.001;
  function =
    "10d+\!\(\*\UnderoverscriptBox[\(\[Sum]\), \(\i = 1\), \
\(\d\)]\)\[\!\(\*\SubsuperscriptBox[\(\x\), \(\i\), \
\(\2\)]\)\-10\cos(2\!\(\*\SubscriptBox[\(\[Pi]x\), \(\i\)]\)\))]";
  interval =
    "x \[Epsilon] <" <> ToString@min <> " ; " <> ToString@max <>
">";
  minValue = 0; params = 0;
  {min, max, interval, function, minValue, params,
RastriginFunction,
  step}
];

functions = {
SphereFunctionPrint -> "Sphere function",
RastriginFunctionPrint -> "Rastrigin function"
};

```

Dále bude definovaná pomocná funkce, která nebude zobrazena v této práci, ale je k dispozici v elektronické příloze do práce.

10. Obsahem dalšího slidu bude definice funkce Manipulate:

- “Section”: “Manipulate”
- “Input”:

```
Manipulate[
  Row[{
    Column[{FunctionPrint[Dimensions, Type][[1]],
      FunctionPrint[Dimensions, Type][[2]],
      FunctionPrint[Dimensions, Type][[3]]}],
    Column[{FunctionPrint[Dimensions, Type][[4]]}],
    Spacer[20]], {Type, functions,
  ControlType -> PopupMenu}, {Dimensions, {2 -> "2D", 3 -> "3D",
  4 -> "4D"}}]
```

Na Obrázek 14 je vidět výsledek funkce Manipulate.

The screenshot shows a Mathematica presentation slide titled "Manipulate". The input code is:

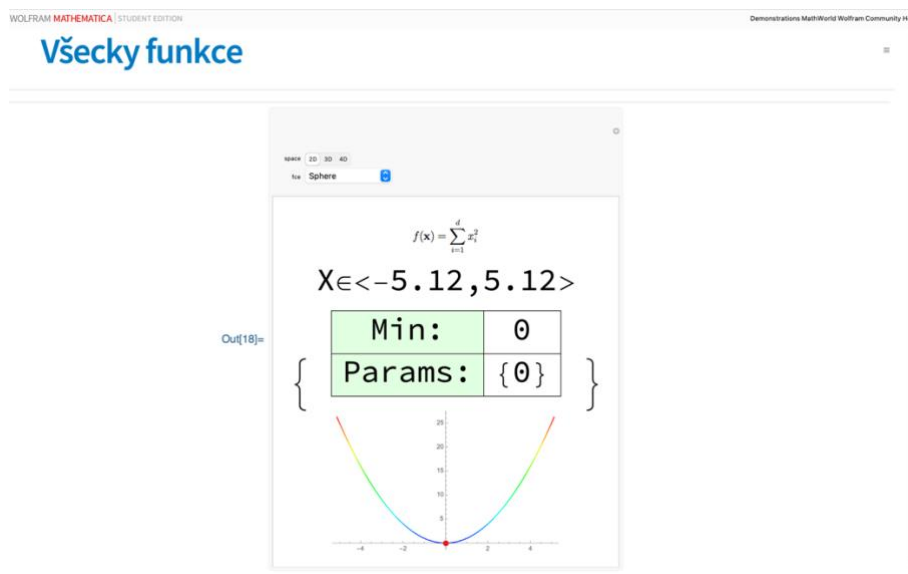
```
In[14]:= Manipulate[
  Row[{
    Column[{FunctionPrint[Dimensions, Type][[1]],
      FunctionPrint[Dimensions, Type][[2]], FunctionPrint[Dimensions, Type][[3]]}],
    Column[{FunctionPrint[Dimensions, Type][[4]]}], Spacer[20]],
  {Type, functions, ControlType -> PopupMenu},
  {Dimensions, {2 -> "2D", 3 -> "3D", 4 -> "4D"}}]
```

The output (Out[14]=) displays:

- A plot of a parabola with the equation  $x \in \langle -5.12 ; 5.12 \rangle$ .
- A table with two rows: "Min:" and "Params:", both with a value of 0.
- A summation formula  $\sum_{i=1}^d x^2$ .

Obrázek 14. Vzhled slidu s funkcí Manipulate

- Na posledním slidu lze ukázat použití funkcí Manipulate pro jejich zobrazení (definice funkcí bude skryta). Skryjeme buňky kódu pomocí nabídky Cell => Cell Properties => vypneme Open a zapneme Initialization Cell. Celý kód prezentace bude umístěn do elektronické přílohy. Na dalším obrázku (viz. Obrázek 15) je zobrazen poslední slid prezentace.



Obrázek 15. Zobrazení všech funkcí pomocí Manipulate

12. Pro potřeby změny stylu, zarovnání obsahu slajdu, změny velikost textu apod. je lepší zahájit prezentaci kliknutím na „Start Presentation“, a při výběru konkrétních oblastí, které chceme změnit, pracujte s „Style Palette“.

## 6 JUPYTER NOTEBOOK

### 6.1 Instalace Jupyter Notebook

Nejjednodušší způsob, jak začít s notebooky Jupyter, je instalace Anaconda, Anaconda je nejpoužívanější distribucí Pythonu pro datovou analýzu a je předinstalovaná se všemi nejoblíbenějšími knihovnamy a nástroji. Kromě Jupyteru patří mezi největší knihovny Pythonu dodávané s Anacondou NumPy, panda a Matplotlib, ačkoli úplný seznam 1000+ je vyčerpávající. To umožní v nejkratší době začít pracovat bez potíží se správou bezpočtu instalací nebo starostí se závislostmi a problémy souvisejícími s OS.

Pro instalaci Anaconda, je potřeba pouze:

1. Stáhnout si nejnovější verzi Anaconda pro Python 3 (ignorujte Python 2.7).
2. Nainstalovat Anaconda podle pokynů na stránce pro stahování nebo ve spustitelném souboru.

Pro pokročilejší uživatele, s již nainstalovaným Pythonem, kteří dávají přednost ruční správě balíčků jednodušeji, bude použit pip [13]:

```
pip3 install jupyter
```

### 6.2 Konfigurace vlastního víceuživatelského serveru JupyterHub (v Docker)

Aby bylo studentům poskytnuto bezproblémové zpřístupnění vzdělávacích materiálů, atraktivní prezentace nových materiálů a navázání spolupráce mezi učitelem a studenty a studenty navzájem je při distanční výuce ideálním řešením vytvořit víceuživatelsky školní server.

Pro konfiguraci základní verze v Docker serveru, je možné použít následující pokyny:

1. Spuštění Docker kontejneru „jhubcontainer“ (který se stáhne z internetu při prvním spuštění) a přepnutí do jeho ovládaní.

```
docker run -it -p 8000:8000 --name jhubcontainer jupyterhub/jupyterhub bash
```

2. Aktualizace seznamu balíčků a jejich závislostí

```
apt update
```

3. Instalace balíčku nezbytných pro fungování JupyterHub

```
apt-get install npm nodejs python3 python3-pip git nano
```

4. Instalace JupyterHub a Jupyter Notebook

```
python3 -m pip install jupyterhub notebook
```

5. Instalace konfigurovatelného proxy serveru

```
npm install -g configurable-http-proxy
```

6. Instalace nativního autentifikačního nástroje

```
cd /home
git clone https://github.com/jupyterhub/nativeauthenticator.git
cd nativeauthenticator
cd nativeauthenticator
pip3 install -e .
```

7. Generování konfiguračního souboru pro JupyterHub ve složce

```
/etc/jupyterhub
mkdir /etc/jupyterhub
cd /etc/jupyterhub
jupyterhub --generate-config -f jupyterhub_config.py
```

8. Otevření konfiguračního souboru v editoru „Nano“

```
nano jupyterhub_config.py
```

a přidání funkce pro správné přesměrování samotných uživatelů do virtuálních serverů, které budou vytvořené díky JupyterHub:

```
import pwd, subprocess
c.JupyterHub.authenticator_class =
'nativeauthenticator.NativeAuthenticator'
c.Authenticator.admin_users = {'admin'}
def pre_spawn_hook(spawner):
    username = spawner.user.name
    try:
        pwd.getpwnam(username)
    except KeyError:
        subprocess.check_call(['useradd', '-ms', '/bin/bash', username])
c.Spawner.pre_spawn_hook = pre_spawn_hook
c.Spawner.default_url = '/notebooks'
```

9. JupyterHub s těmito nastaveními se spouští pomocí příkazu

```
jupyterhub -f /etc/jupyterhub/jupyterhub_config.py
```



10. Pokud Docker probíhá lokálně na počítači, adresa JupyterHub by měla být:

<http://localhost:8000>

11. Zpočátku je třeba vytvořit nového uživatele se jménem “admin” a libovolným heslem. Pak je možné se přihlásit s těmito údaji:

12. Noví uživatelé se přidávají na adrese /hub/authorize, kde se taky schvalují žádosti nových uživatelů, kteří si chtějí založit účet a čekají na její schválení.

<http://localhost:8000/hub/authorize>

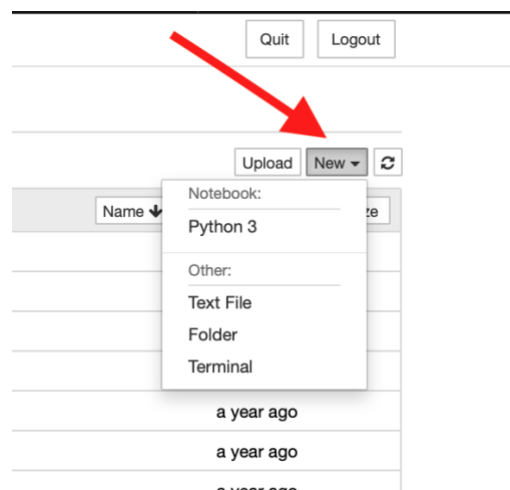
### 6.3 Jak vytvořit prezentační dokument v Jupyter Notebooku

1. Zpočátku je potřeba spustit Jupyter Notebook pomocí příkazu v příkazovém řádku:

```
jupyter notebook
```

Webová stránka Jupyter IDE by se měla otevřít automaticky. Pakliže se tak nestane, ve výpisu v příkazovém řádku bude vypsána adresa, na které je spuštěn Jupyter Notebook.

2. Dále je nutno vytvořit klasický notebook Jupyter, kliknutím na New => Python 3 (viz. Obrázek 16).



Obrázek 16. Menu pro vytvoření nového dokumentu v Jupyter Notebook

3. Pro změnu formátu zobrazení dokumentu ve formě prezentace je nutné provést následující kroky: View => Cell Toolbar => Slideshow.

4. Po změně režimu zobrazení na prezentaci, v pravém horním rohu každé buňky, by se měla zobrazit rozbalovací nabídka pro výběr typu buňky (snímek, dílčí snímek, fragment, přeskočit nebo poznámka).

Pomocí této nabídky můžeme vytvářet běžné snímky, dílčí snímky. Typ poznámky nám pomůže skrýt tento blok před prezentací. Pokud jej nechceme ukázat, je nutné, aby prezentace fungovala správně.

5. Pro finální vytvoření prezentace z našeho poznámkového bloku ve formátu Prezentace je třeba spustit následující příkaz v příkazovém řádku (ve složce, kde se nachází dokument, který chceme exportovat do prezentace):

```
jupyter nbconvert Jupyter\ Slides.ipynb --to slides --post serve
```

6. Výsledkem bude, že ve složce s poznámkovým blokem, kde je vytvořena prezentace, ale stále má příponu .pynb, bude vytvořena stránka HTML, vygenerovaná z poznámkového bloku. Po spuštění daného příkazu se automaticky spustí lokální server zobrazující prezentaci, který se otevře v novém okně.

Mezi slidy je možné přepínat pomocí tlačítek v pravém dolním rohu nebo kláves se šipkami na klávesnici. Pro přepínání tam a zpět, je třeba použít klávesy vlevo x vpravo, pro pohyb na dílčí slidy klávesou dolů x nahoru.

Pokud v prezentaci byly bloky označené jako poznámka, je vidět, že se v prezentaci nezobrazily [14].

### 6.3.1 Interaktivita v Jupyter

Export notebooku do prezentace ve formátu HTML má jednu zásadní nevýhodu – není dynamická. Pokud jsou použité interaktivní prvky (tlačítka, dropdown menu apod.), tyto nebudou aktivní. Pro přidání prezentací v interaktivitě, můžeme použít Reveal.js - rozšíření Jupyter / IPython Slideshow (RISE). Jedná se o plugin, který oživí prezentace a přidá možnost spouštět kód napsaný v notebooku přímo z prezentace, avšak prezentace se bude zobrazovat trochu jiným způsobem (přímo v prostředí Jupyter).

Interaktivní prezentace je dynamická, pouze pokud je otevřena z běžícího poznámkového bloku pomocí tlačítka, které by mělo být v nabídce po instalaci RISE.

Instalace a zpřístupnění RISE v Jupyter Notebook [15]:

```
pip install RISE
jupyter-nbextension install --py --sys-prefix
jupyter nbextension enable --py --sys-prefix
pip install ipywidgety
```

### 6.3.2 Příklad vytvoření prezentačního dokumentu v Jupyteru pro předmět matematika

1. Nejprve je potřeba spustit Jupyter Notebook a vytvořit nový soubor.
2. Dále je nutno změnit formát prezentace: View => Cell Toolbar => Slideshow.
3. Pro vytvoření nadpisu je třeba změnit formát buňky (viz. Obrázek 17 nebo Obrázek 18) na Markdown a vložit následující kód HTML:

```
<center><h1>Testování heuristik</h1></center>
```



Obrázek 17. Menu pro změnu typu buňky ("Kód")



Obrázek 18. Menu pro změnu typu buňky ("Poznámka")

(pro usnadnění stylování textových bloků lze používat prvky HTML nebo LaTeX)

4. Dále je třeba poznamenat, že tato buňka je slide (jinak se v prezentaci nezobrazí). Pro tyto úpravy je nutné v pravém rohu buňky vybrat: Slide Type => Slide.
5. Pro tvorbu nového slidu je nutné vytvořit novou buňku (nevytváříme slide, ale vytváříme buňky, které označíme jako snímek, dílčí snímek, fragment, poznámku, přeskočit). Tento slide bude obsahovat textové informace, proto je třeba změnit jeho typ na Markdown a vložit následující kód:

```
<h2>Testování heuristických algoritmů</h2>
<ul>
  <li>
    Pro testování optimalizačních algoritmů se používají dva rozdílné
    postupy
  </li>
  <li>
    V prvním obvykle vycházíme z již existujících příkladů, které již
    byly řešeny jinými algoritmy. Výsledky právě testovaného algoritmu pak
    porovnáme s výsledky již existujícími (publikovanými v literatuře)
  </li>
  <li>
    Druhý způsob spočívá v tom, že použijeme speciálně sestavenou množinu
    testovacích funkcí (tzv. benchmark set nebo také testbed), obsahující
    funkce s různými vlastnostmi, kombinace unimodálních, multimodálních,
    hybridních a kompozitních funkcí
  </li>
</ul>
```

Na Obrázek 19 můžeme vidět vzhled daného slidu, který byl vytvořen pomocí postupu uvedeného výše.



## Testování heuristických algoritmů

- Pro testování optimalizačních algoritmů se používají dva rozdílné postupy
- V prvním obvykle vycházíme z již existujících příkladů, které již byly řešeny jinými algoritmy. Výsledky právě testovaného algoritmu pak porovnáme s výsledky již existujícími (publikovanými v literatuře)
- Druhý způsob spočívá v tom, že použijeme speciálně sestavenou množinu testovacích funkcí (tzv. benchmark set nebo také testbed), obsahující funkce s různými vlastnostmi, kombinace unimodálních, multimodálních, hybridních a kompozitních funkcí



Obrázek 19. Vzhled slidu vytvořeného pomocí Jupyter notebook slides.

6. Přidáním další buňky typu slide vytvoříme nový slide. Formát buňky bude Markdown a obsahem následující text:

```
<h2>Testování heuristických algoritmů</h2>
<ul>
  <li>
    Vzhledem k tomu, že jsou známy analytické vztahy, je u většiny z nich
    velmi jednoduché vypočítat pozici a hodnotu extrému pro libovolnou dimenzi
  </li>
  <li>
    Pro profesionální testování se používají testovací sady například pro
    IEEE CEC Competitions nebo GECCO kongres
  </li>
</ul>
```

7. V následujícím návodu je použita knihovna PyBenchFCN<sup>11</sup>, která obsahuje definice různých funkcí pro testování heuristických algoritmů a nástroje pro vykreslování jejich grafů. Pro jejich použití je potřeba zpočátku nainstalovat do počítače pomocí:

```
pip install PyBenchFCN
```

8. Na novém slidu naimportujeme potřebné knihovny, definice seznamu názvů funkcí pro jejich další použití pro výběr z dropdown-menu a definujeme funkci pro skrytí programového kódu z prezentace, které se dá ovládat pomocí tlačítka. Takovým způsobem můžeme docílit zobrazení kódu, tam, kde je to potřebné a jeho skrytí, pokud jeho zobrazení není potřebné. Obsahem tohoto slidu (formátu kódu) bude následující kód:

---

<sup>11</sup> <https://github.com/Y1fanHE/PyBenchFCN>

```

import numpy as np

from PyBenchFCN import SingleObjectiveProblem as SOP
from PyBenchFCN import Tool
import ipywidgets as widgets
n_var = 10 # dimension of problem
n_pop = 3 # size of population
problem = SOP.ackleyfcn(n_var) # Ackley problem
xl, xu = problem.boundaries # bound of problem
x = np.random.uniform(xl, xu, n_var) # initialize a solution
X = np.random.uniform( xl, xu, (n_pop, n_var) ) # initialize a population
import ipywidgets as widgets
from IPython.display import display, HTML
javascript_functions = {False: "hide()", True: "show()"}
button_descriptions = {False: "Show code", True: "Hide code"}
def toggle_code(state):
    output_string = "<script>$(\"div.input\").{}/script>"
    output_args = (javascript_functions[state],)
    output = output_string.format(*output_args)
    display(HTML(output))
def button_action(value):
    state = value.new
    toggle_code(state)
    value.owner.description = button_descriptions[state]
state = True
toggle_code(state)
button = widgets.ToggleButton(state, description =
button_descriptions[state])
button.observe(button_action, "value")
functions = [
'Ackley', 'Adjiman',
'Bartelsconn', 'Beale', 'Bird', 'Booth', 'Brent', 'Brown',
'Dropwave', 'Easom', 'Eggholder', 'Exponential',
'Griewank', 'Himmelblau', 'Keane', 'Leon',
'Matyas', 'McCormick', 'Periodic', 'Picheny', 'Qing', 'Quartic', 'Rastrigin',
'Ridge', 'Rosenbrock', 'Salomon',
'Schwefel', 'Sphere', 'Styblinski',
'Zakharov'
]
display(button)

```

“Display (button)” je prvek pro ovládání zobrazení programového kódu. Můžeme ho zkopírovat na každý slide, kde je potřeba možnost skrýt nebo otevřít programový kód [16].

9. Na dalším slidu bude dropdown menu pro výběr funkce k jejímu zobrazení a samotnému vykreslení. Obsahem je následující kód:

```
dropdown = widgets.Dropdown(
    options=functions,
    value='Ackley',
    description='Function:',
    disabled=False,
)
output = widgets.Output(layout={'border': '1px solid black', 'width':
'300px'})

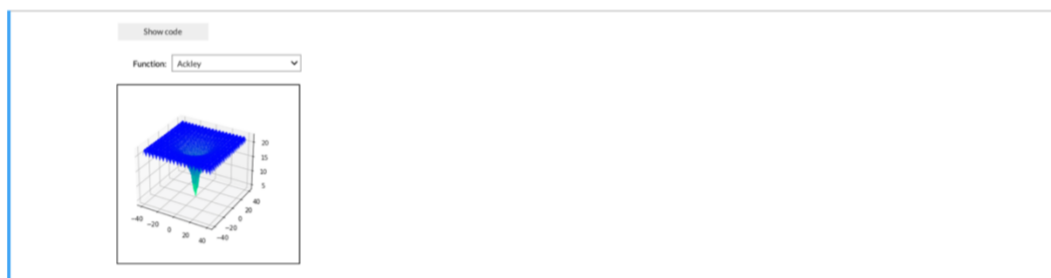
display(button, dropdown, output)
def funcPlot(change):
    with output:
        output.clear_output()
        Tool.plot_sop(change['new'].lower())
dropdown.observe(funcPlot, names='value')
with output:
    Tool.plot_sop('ackley')
```

10. Prezentace je tímto hotova. Pro její správné spuštění, je třeba zpočátku spustit každou buňku pomocí Shift + Enter nebo přes menu Cell => Run All. Poté, když spustíme všechny buňky (slidy), lze spustit prezentaci pomocí tlačítka, které je zobrazeno na Obrázek 20.



Obrázek 20. Tlačítko pro spuštění prezentace

Dále je zobrazen poslední slide prezentace (viz. Obrázek 21).



Obrázek 21. Poslední slide prezentace vytvořené pomocí Jupyter notebook slides



## ZÁVĚR

Dnes lze jednu z nejběžnějších forem poskytování výukových materiálů nazvat multimediální prezentace. Instruktažní prezentace představují pohodlný a efektivní způsob prezentace informací pomocí počítačových programů (například Microsoft PowerPoint, Apple Keynote) a webových služeb (Prezi, Google Slides atd.). K dosažení tohoto cíle se používají následující prvky prezentace: text, obrázek, zvuk, video, animace, interaktivita. Používání multimediálních prezentací ve výukovém procesu mění povahu tradiční učebny, takže je živější a zajímavější. Využití multimédií ve třídě pomáhá rozšířit obecný rozhled studentů a obohacuje jejich znalosti.

Na začátku teoretické části je popsána role dynamických prezentací při asimilaci informací studenty. Následující text popisuje technologie Jupyter notebook slides a Mathematica slideshow notebook, jejich charakteristické vlastnosti, základy programovacích jazyků Wolfram Language a Python. Rovněž jsou v teoretické části popsány možnosti spouštění notebooků Jupyter, na jejichž základě lze s jistotou říci, že technologie Jupyter notebook slides je vhodnější pro distanční vzdělávání a určitou spolupráci. Prezentaci, vytvořené v Mathematica slideshow notebook, lze otevřít pouze v prostředí Mathematica, nainstalované na počítači, bez možnosti pracovat na jednom dokumentu pro více lidí. Jupyter naopak umožňuje spouštění poznámkových bloků (prezentací) různými způsoby na serveru, což usnadňuje činnost studentům, kteří nemají možnost koupit a nainstalovat prostředí Mathematica s přístupem k nim.

Praktická část popisuje, jak nainstalovat Mathematica a jak pomocí této technologie vytvářet prezentace a dále podrobný návod k vytvoření prezentace pro předmět Informační matematika. Další kapitola popisuje, jak nainstalovat a spustit prostředí Jupyter na osobním počítači a jak spustit server JupyterHub pro více uživatelů v Dockeru. Je třeba poznamenat, že po dokončení konfigurace školního serveru podle pokynů uvedených v této práci získají noví uživatelé (studenti) přístup k výpočetnímu výkonu až poté, co jejich žádost přijme správce (učitel). Otázka zapouzdření uživatelských dat zůstala nevyřešena. Všechna data, ke kterým má Jupyter přístup, budou k dispozici všem uživatelům, čehož lze dosáhnout prací s jejich nastavením. V případě potřeby používat prezentace, vytvořené pomocí Jupyter notebook slides, postačí lokální instalace na jejich počítači.

V následujících kapitolách je popsán základ pro vytvoření prezentace pomocí technologie Jupyter notebook slides a podrobný návod, jak vytvořit prezentaci pro předmět Informační matematika.

Při psaní práce vyšlo najevo, že technologie Jupyter notebook slides, je vhodnější pro distanční vzdělávání. Nabízí více způsobů distribuce prezentací na internetu a rovněž příležitost, kdy pokud od této práce odbočíme, je možné pomocí funkce CoCalc poskytnout vysokou úroveň spolupráce při práci na jednom dokumentu velkému počtu studentů současně. Mathematica slideshow notebook naopak neposkytuje možnost publikovat prezentace nebo notebooky na internetu, tak aby se s nimi dalo pracovat, aniž by se musel nainstalovat Wolfram Mathematica, případně možnost současné práce na jednom dokumentu skupinou studentů. Na druhou stranu notebook Mathematica slideshow nabízí více nástrojů pro vytváření krásných a dynamických prezentací, které v předmětech, kde je nutné předvést práci s kódem, vykreslení funkcí nebo jakoukoli jinou dynamikou, budou ideálním kandidátem pro nahrazení standardních prezentací, které jsou již dnešní normou. Jazyk Wolfram je navíc ideální pro matematiky.

**SEZNAM POUŽITÉ LITERATURY**

- [1] *Wolfram Mathematica* [online]. [cit. 2021-5-17]. Dostupné z:  
<https://www.wolfram.com/mathematica/>
- [2] *Wolfram Mathematica Documentation* [online]. [cit. 2021-5-17]. Dostupné z:  
<https://reference.wolfram.com/language/>
- [3] *Wolfram Alpha: Computational Intelligence* [online]. [cit. 2021-5-17]. Dostupné z:  
<http://www.wolframalpha.com/>
- [4] ANANICHEVA, D a I KORYAKOV. *FUNDAMENTALS OF CRYPTOLOGY: A Professional Reference and Interactive Tutorial* [online]. 1. Moskva: Mir, 2006 [cit. 2021-5-17]. ISBN 5-03-003639-3. Dostupné z:  
<https://www.sgu.ru/structure/computersciences/theorcompsafe/osnovnye-uchebniki-i-rekomenduemye-materialy/tilborg-van-h-k-osnovy-kriptologii-professionalnoe>
- [5] MEDVEDEV, Anton. *Úvod do Wolfram Mathematica* [online]. In: 25.05.2013 [cit. 2021-5-17]. Dostupné z: <https://habr.com/ru/post/180925/>
- [6] Wolfram (jazyk programování). *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-5-17]. Dostupné z: [https://ru.wikipedia.org/wiki/Wolfram\\_\(язык\\_программирования\)](https://ru.wikipedia.org/wiki/Wolfram_(язык_программирования))
- [7] BAJPAYEE, Abhishek. *5 Reasons Why Jupyter Notebooks are Perfect for Teaching* [online]. 2017 [cit. 2021-5-17]. Dostupné z: <https://medium.com/gryd-notebooks/5-reasons-why-jupyter-notebooks-are-perfect-for-teaching-24c6e99ca150>
- [8] *Teaching and Learning with Jupyter: Chapter 6 Getting your class going with Jupyter* [online]. [cit. 2021-5-17]. Dostupné z:  
<https://jupyter4edu.github.io/jupyter-edu-book/getting-going.html>
- [9] Voila-Dashboards. *GitHub* [online]. [cit. 2021-5-17]. Dostupné z:  
<https://github.com/voila-dashboards/voila>
- [10] MCHOST. *Chto takoe Python* [online]. 07.12.2020 [cit. 2021-5-17]. Dostupné z:  
<https://mchost.ru/articles/chto-takoe-python/>

- [11] AHADJANOV, Maksim. *Dlya chego nuzhen yazyk Python* [online]. 03.06.2020 [cit. 2021-5-17]. Dostupné z: <https://gb.ru/posts/dlya-chego-nuzhen-yazyk-python>
- [12] ITSUMA. *V nachale etogo goda Python smestil Java i stal vtorym po populyarnosti yazykom programmirovaniya sredi razrabotchikov* [online]. 03.03.2020 [cit. 2021-5-17]. Dostupné z: <https://habr.com/ru/company/itsumma/news/t/490834/>
- [13] PRYKE, Benjamin. Jupyter Notebook for Beginners: A Tutorial. *Towardsdatascience* [online]. 09.05.2018 [cit. 2021-5-17]. Dostupné z: <https://towardsdatascience.com/jupyter-notebook-for-beginners-a-tutorial-f55b57c23ada>
- [14] SPECK, Matthew. Presenting Code Using Jupyter Notebook Slides. *Medium* [online]. 16.05.2017 [cit. 2021-5-17]. Dostupné z: <https://medium.com/@mjspeck/presenting-code-using-jupyter-notebook-slides-a8a3c3b59d67>
- [15] MIKE. Creating Presentations with Jupyter Notebook. *Mouse vs Python* [online]. 25.09.2018 [cit. 2021-5-17]. Dostupné z: <https://www.blog.pythonlibrary.org/2018/09/25/creating-presentations-with-jupyter-notebook/>
- [16] SHEPHERD, Erick. *Stackoverflow* [online]. 04.11.2018 [cit. 2021-5-17]. Dostupné z: <https://stackoverflow.com/questions/27934885/how-to-hide-code-from-cells-in-ipython-notebook-visualized-with-nbviewer>

**SEZNAM OBRÁZKŮ**

Obrázek 1. Základní matematický asistent .....	13
Obrázek 2. Volba typu nového dokumentu .....	26
Obrázek 3. Základní nastavení prezentačního dokumentu .....	27
Obrázek 4. Menu pro přidání nového prvku slidu .....	28
Obrázek 5. Nabídka prvku pro přidání do slidu.....	28
Obrázek 6. Tlačítko pro přidání nového slidu .....	28
Obrázek 7. Nabídka typu nového slidu.....	29
Obrázek 8. Tlačítko pro spuštění prezentace .....	29
Obrázek 9. Menu pro správu prezentace .....	30
Obrázek 10. Menu pro správu buňky.....	30
Obrázek 11. Záhavní slide prezentace v Mathematica slideshow notebook .....	31
Obrázek 12. Příklad použití funkce Plot ve Wolfram.....	33
Obrázek 13. Příklad použití funkce Plot3D .....	33
Obrázek 14. Vzhled slidu s funkcí Manipulate.....	36
Obrázek 15. Zobrazení všech funkcí pomoci Manipulate .....	37
Obrázek 16. Menu pro vytvoření nového dokumentu v Jupyter Notebook.....	40
Obrázek 17. Menu pro změnu typu buňky ("Kód").....	42
Obrázek 18. Menu pro změnu typu buňky ("Poznámka").....	42
Obrázek 19. Vzhled slidu vytvořeného pomoci Jupyter notebook slides.....	43
Obrázek 20. Tlačítko pro spuštění prezentace .....	46
Obrázek 21. Poslední slide prezentace vytvořené pomocí Jupyter notebook slides...47	

## SEZNAM PŘÍLOH

P I Obsah CD

## **PŘÍLOHA P I: OBSAH CD**

/text – text práce

/mathematica – ukázková prezentace v Mathematica slideshow notebook

/jupyter – ukázková prezentace v Jupyter notebook slides