

Zadanie wprowadzające dla osób prowadzących zajęcia na poziomie „Master”

Szanowni Państwo,

Poniżej znajdziecie Państwo materiał, który wykorzystacie do rozpoczęcia zajęć z uczniami.

Uprzejmie prosimy o zapoznanie się z nim, a następnie o przygotowanie refleksji i uwag, które będą mogli Państwo przedstawić i omówić na forum w trakcie pierwszego dnia szkolenia. Polecamy wykonać ćwiczenia, które znajdują się poniżej.

Warszawa, 24 września 2019 r.

Moduł 5. Wprowadzenie do narzędzia Google Colab

Cele:

- o uczeń potrafi uruchomić kod pythonowy napisany w Google Colab
- o uczeń potrafi zapisać napisany program
- o uczeń potrafi dodać komentarze do treści programu
- o uczeń potrafi napisać prostą funkcję i ją wywołać

Zasoby:

- o komputery z dostępem do Internetu
- o Konto Gmail

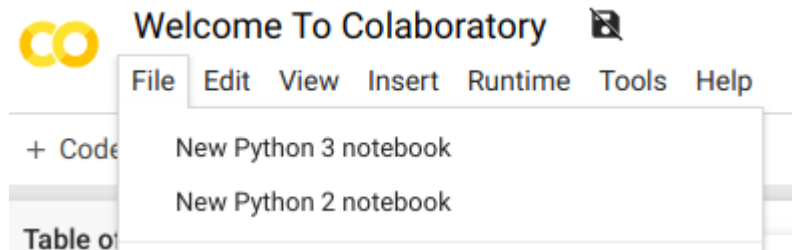
Wprowadzenie:

Nauczyciel zwraca się do uczniów: Zaczynamy właściwą naukę programowania. Będziemy pisać kody w programie Python (wersja 3.6). Kod możemy napisać zarówno na kartce, jak i w notatniku lub innym edytorze tekstu. Jednak aby go uruchomić musimy użyć konsoli bądź interpretera tekstu. Chcielibyśmy uniknąć problemów technicznych, dlatego będziemy używać Google Colab. Dzięki temu będziecie mogli uruchamiać kod praktycznie z każdego narzędzia i nie musicie instalować bibliotek.

Dzisiaj nauczymy się obsługi tego narzędzia. Dowiemy się jak:

- o uruchomić napisany program,
- o zapisać napisany program,
- o otworzyć wcześniej zapisany program,
- o dodać komentarze do treści programu,
- o napisać i wywołać prostą funkcję.

Nauczyciel prosi uczniów: Otwórzcie teraz przeglądarki i wejdźcie na stronę colab.research.google.com. Uruchomcie nowy notatnik wchodząc w zakładkę **File**, następnie **new python3 notebook**.



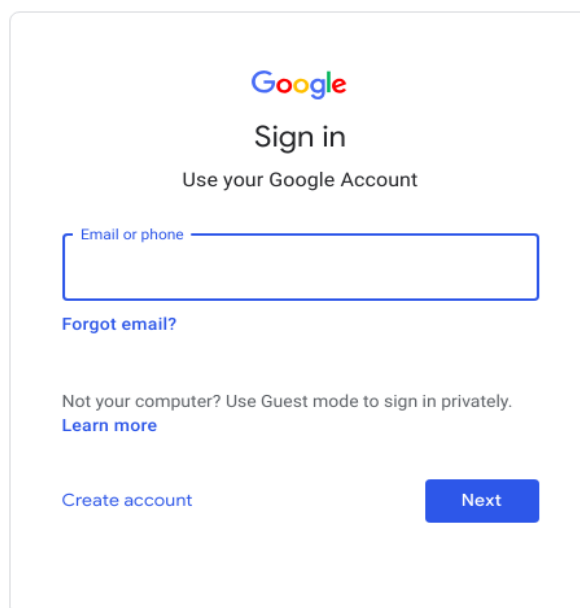
Zostaniecie poproszeni o zalogowanie się do swojego profilu **Gmail**.

Google sign-in required

You must be logged in to a Google account to continue.

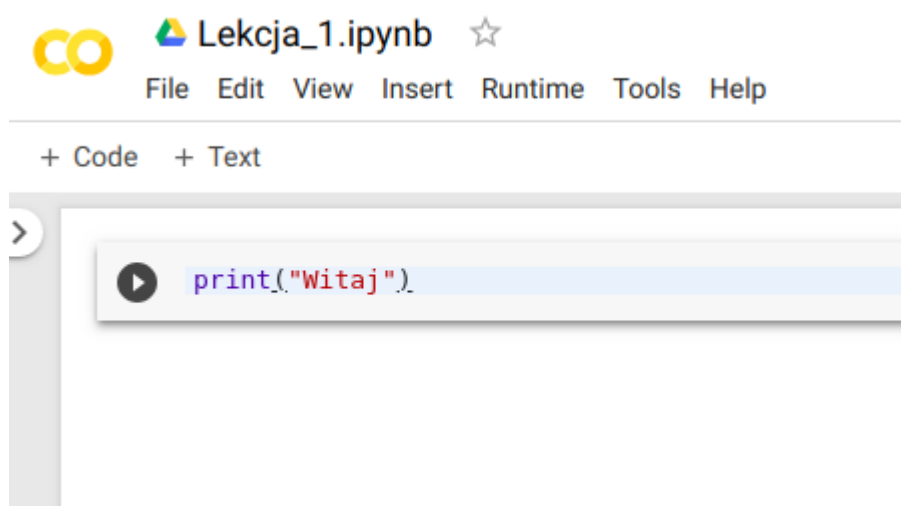
OK

Po kliknięciu w **OK** przejście do okna logowania:

A screenshot of the Google sign-in page. At the top center is the Google logo. Below it is the text 'Sign in' and 'Use your Google Account'. There is a text input field with the placeholder text 'Email or phone'. Below the input field is a link that says 'Forgot email?'. Further down, there is a line of text: 'Not your computer? Use Guest mode to sign in privately.' followed by a link 'Learn more'. At the bottom left is a link 'Create account' and at the bottom right is a blue button labeled 'Next'.

Po zalogowaniu otworzy się notatnik Python, z którego działaniem zapoznamy się wykonując następujące ćwiczenie:

Każdy nowy notebook po otwarciu ma nazwę **Untitled.ipynb**. Warto jednak zmienić nazwę tak, żebyśmy wiedzieli jakie treści się w nim znajdują. W tym celu najeżdżamy kursorem na ten napis i klikamy. Wówczas pole zamieni się w edytowalne i będziemy mogli wpisać dowolną nazwę. Pamiętajcie, że plik musi mieć rozszerzenie **pynb**, możemy nazwać go np. **Lekcja_1.ipynb**. Napiszemy prosty program, którego zadaniem będzie wypisanie na ekran słów: **Witaj**. Wpiszcie teraz w oknie tekst: **print("Witaj")**.



Program możemy uruchomić na dwa sposoby: bezpośrednio z klawiatury naciskając jednocześnie **Ctrl + Enter** lub klikając kursorem myszy w biały znak strzałki znajdującym się w szarym kółku na początku każdego pola tekstowego. Żeby napisać kolejny kawałek programu możemy kontynuować pisanie w kolejnych liniach pierwszego okna z kodem, lub kontynuować w nowym polu. **Nauczyciel pyta uczniów:** Jaka jest różnica pomiędzy tymi dwiema możliwościami?

Odpowiedź: Pisząc w nowych oknach możemy uruchamiać program częściami. Dzięki temu będziemy mieli możliwość sprawdzania i wypisywania na ekran częściowych wyników programu. Dodatkowo jeżeli któraś część trwa długo, to wystarczy uruchomić ją tylko jeden raz i potem można uruchamiać kolejne części programu.

Aby otworzyć nowe pole z kodem klikamy na pole **+ Code** znajdujące się tuż nad edytorem tekstu. W nowym polu możemy napisać kod który wypisze nam na ekran cyfrę 5. Możemy zrobić to bezpośrednio za pomocą komendy **print(1)**, bądź poprzez stworzenie nowej zmiennej **x** i przypisanie jej wartości **x=1**, a następnie wypisać zmienną **print(x)**.

```
[3] print (1)
```

↳ 1

```
▶ x=1  
print (x)
```

↳ 1

Zauważmy, że aby wypisać na ekran tekst, musimy go oznaczyć cudzysłowem, gdy chcemy wypisać liczbę, lub dowiedzieć się co kryje się pod daną zmienną nie musimy nic więcej dodawać.

Wśród dobrych praktyk programistycznych znajdziemy zasadę code review. Czyli kod napisany przez jednego programistę jest sprawdzany i ulepszany, przez drugiego. Wówczas użyteczne jest robienie dla siebie pythonowym. W celu dodania komentarza

```
[6] x=1 przypisanie zmiennej x wartości 1  
print (x)
```

↳ File "<ipython-input-6-ff009b29453e>", line 1
x=1 przypisanie zmiennej x wartości 1
^
SyntaxError: invalid syntax

SEARCH STACK OVERFLOW

```
▶ x=1 #przypisanie zmiennej x wartości 1  
print (x)
```

↳ 1

przed tekstem dajemy znak **#** wówczas komputer nie próbuje czytać ani wykonywać komend występujących po tym znaku. Widać to na przykładzie poniżej:

Jeżeli chcemy zakomentować kilka linii kodu możemy je wziąć w potrójny cudzysłów **"""**. W notatniku zakomentowany tekst będzie w kolorze czerwonym.

```
[8] x=1 #przypisanie zmiennej x wartości 1
    komputerze, nie czytaj tego
    to tylko moje komentarze
    print (x)
```

File "<ipython-input-8-c087cc50c1d6>", line 2
komputerze, nie czytaj tego
^
SyntaxError: invalid syntax

SEARCH STACK OVERFLOW

```
x=1 #przypisanie zmiennej x wartości 1
'''komputerze, nie czytaj tego
to tylko moje komentarze'''
print (x)
```

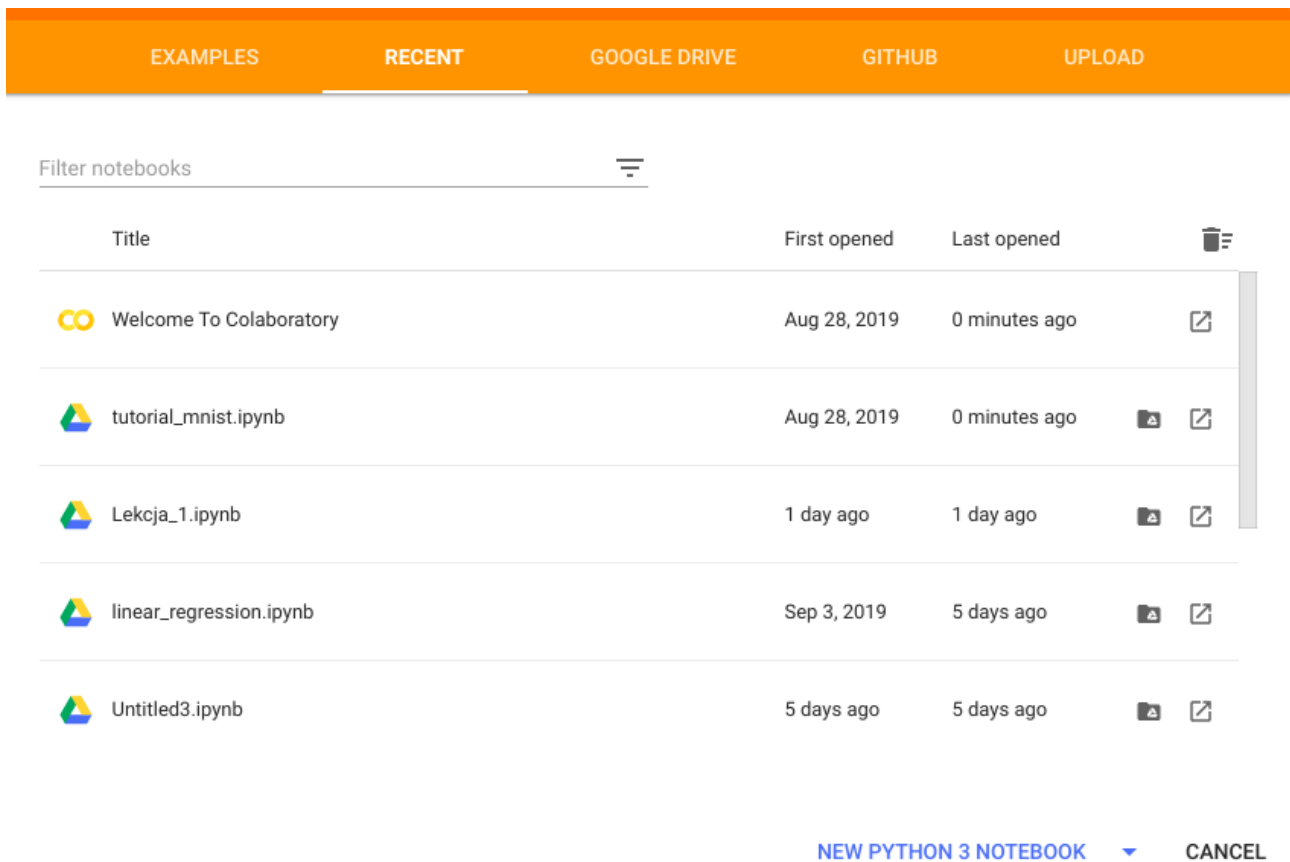
1

Gdy nie oznaczyliśmy naszych notatek jako komentarz, program nie może się wykonać, bo odczytuje te linie jako błędne. Wówczas wyświetla **SyntaxError: invalid syntax**. Oznacza to błąd składni języka Python.

Gdy skończymy edytować plik możemy zapisać go na swoim dysku. W tym celu klikamy w pole **File**, a następnie **Save a copy on a drive**. Po zapisaniu pliku możemy wyjść z naszego edytora. Zamykając zakładkę w wyszukiwarce. Aby otworzyć zapisany już notatnik wchodzimy w link: <https://colab.research.google.com/drive> i zostaniemy przekierowani do strony:

<https://colab.research.google.com/notebooks/welcome.ipynb#recent=true>.

Na ekranie pojawi się okno:



The screenshot shows the Google Colab interface with the 'RECENT' tab selected. At the top, there are navigation tabs: 'EXAMPLES', 'RECENT', 'GOOGLE DRIVE', 'GITHUB', and 'UPLOAD'. Below the tabs is a search bar labeled 'Filter notebooks' and a hamburger menu icon. The main area displays a list of notebooks with the following columns: 'Title', 'First opened', 'Last opened', and a set of icons (trash, share, refresh). The notebooks listed are:

Title	First opened	Last opened	Icons
Welcome To Colaboratory	Aug 28, 2019	0 minutes ago	Share, Refresh
tutorial_mnist.ipynb	Aug 28, 2019	0 minutes ago	Trash, Share, Refresh
Lekcja_1.ipynb	1 day ago	1 day ago	Trash, Share, Refresh
linear_regression.ipynb	Sep 3, 2019	5 days ago	Trash, Share, Refresh
Untitled3.ipynb	5 days ago	5 days ago	Trash, Share, Refresh

At the bottom right of the interface, there are two buttons: 'NEW PYTHON 3 NOTEBOOK' and 'CANCEL'.

Możemy teraz otworzyć nasz plik klikając w niego prawym kursorem myszy. Możemy go teraz edytować i rozwijać. Na kolejnej lekcji będziemy mogli do niego wrócić.

Nauczyciel mówi – Kolejnym waszym zadaniem będzie napisanie w otwartym już pliku *Lekcja_1.ipynb* kilku funkcji: np. dodawanie, mnożenie, dzielenie. Poniżej przedstawiamy jak napisać funkcję dodawanie.

```
def dodawanie(x,y):
    return(x+y)

print(dodawanie(5,6))
```

11

Definicja funkcji musi zawierać argumenty t.j.: **xiy,i** kończyć się dwukropkiem. Treść zaś nie zaczyna się od początku linii, tylko po tabulacji. W notatniku Colab dzieje się

to automatycznie. Napiszcie teraz funkcji odejmowanie i mnożenie, które zwrócą wyniki powyższych działań. Dodatkowo możecie też stworzyć funkcję, która będzie rozwiązywała zadania z fizyki.

Nauczyciel prosi uczniów – Stwórzcie teraz program, który rozwiąże obliczy jaką drogę przebył samochód, który poruszał się z prędkością 40 km/h. Następnie przez 6 minut poruszał się ruchem jednostajnie przyspieszonym z przyspieszeniem równym 100 km/h². Jaką drogę przebył w tym czasie?

Przypomnijmy, że drogę w ruchu jednostajnie przyspieszonym obliczamy ze wzoru

$s = v_0 * t + 0,5 * a * t * t$, gdzie:

- o v_0 – prędkość początkowa
- o a – przyspieszenie
- o t – czas trwania ruchu

```
[6] #program przyjmuje prędkość w km/h, przyspieszenie w km/h^2, i czas w godzinach
def droga(v_0, a, t):
    s = v_0*t + 0.5 * a* t*t
    return (s)

print(droga(v_0 = 40, a=100, t=0.1))
```

↳ 4.5

Na dzisiejszych zajęciach poznaliśmy narzędzie do tworzenia i edycji kodu w języku python. Umiemy utworzyć i zapisać pliki. Dodatkowo możemy napisać proste funkcje wykonujące działania arytmetyczne i je wywołać.