



Politechnika Wrocławska

Programowanie Obiektowe **Wykład #1 (W#15)**



HR EXCELLENCE IN RESEARCH

Janusz Andrzejewski



Wstęp — „Programowanie obiektowe”

- Informacje organizacyjne
- Programowanie — dlaczego i co to jest?
- Dlaczego Python?
- Python: historia, cechy, filozofia
- Czego będziemy się uczyli?
- Cel zajęć
- Literatura i inne pomoce w nauce
- Python:
 - Elementy programu
 - Budowa programu

Kontakt:

- janusz.andrzejewski@pwr.edu.pl
- Pokój 319, budynek A1
- Prezentacje są i będą dostępne na:
<https://eportal.pwr.edu.pl/course/view.php?id=1238>



Kiedy:

- Wykłady w czwartki (nieparzyste) o godz. **13.15** — sala **A2** bud. **L1**
- Laboratoria w środy **11.15 249A1**
 - tygodnie nieparzyste
 - tygodnie parzyste – dr. inż. Karol Kulinowski

	PAŹDZIERNIK						LISTOPAD				GRUDZIEŃ				STYCZEŃ					LUTY		
PN	25	2	9 Pn/P	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	5	12	19
WT	26	3	10	17	24	31	7	14	21	28	5	12 Śr/P	19	26	2	9	16	23	30 Wt/P	6	13	20
ŚR	27	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	31 Śr/P	7	14	21
CZ	28	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	1 Cz/P	8	15	22
PT	29	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	2 Pt/P	9	16	23
SO	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	3	10	17	24
N	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25
P - PARZYSTY N - NIEPARZYSTY	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P

■ Święta
 ■ Dodatkowe dni wolne
 ■ Zmiany
 ■ Sesja
 ■ Przerwa międzysemestralna

Zaliczenie:

- Zaliczenie wykładu — spełnienie wszystkich warunków:
 - Zaliczone zajęcia laboratoryjne
 - Zaliczone kolokwium na ostatnim wykładzie:
 - Kolokwium w formie testu wielokrotnego wyboru
 - Pytania będą zarówno „teoretyczne” jak i „praktyczne”
 - Ilość poprawnych odpowiedzi nie jest stała
 - Przy jednej odpowiedzi nie ma punktów ujemnych za błędną odpowiedź
 - Punkty ujemne są przydzielane od dwu lub więcej udzielonych odpowiedzi (poprawnych lub błędnych)
- Ocena końcowa:
 - Średnia ocen z laboratorium oraz wykładu

Dlaczego programowanie:

- Programowanie uczy myślenia
- Programowanie rozwija kreatywność
- Programowanie uczy posługiwania się zasadami
- Programowanie to umiejętność, na którą rośnie zapotrzebowanie
- Programowanie jest interdyscyplinarne — wykorzystuje zarówno wiedzę ścisłą jak i humanistyczną
- Umiejętność programowania staje niezbędną dodatkową wiedzą oprócz podstawowego wykształcenia inżyniera

Co to jest programowanie?

- 1) **Programowanie** — mówienie(rozkazywanie) w sposób zrozumiały komputerowi dokładnie i precyzyjnie tego co ma robić
- 2) Co znaczy że umiem programować
 - 1) Znam (umiem poszukać lub wymyślić) i rozumiem algorytm — sposób rozwiązania problemu
 - 2) Znam jakiś język programowania
 - 3) Umiem zaimplementować algorytm wykorzystując cechy wybranego języka programowania
 - 4) Umiem uruchomić (skompilować, usunąć błędy składniowe – syntaktyczne) napisany program
 - 5) **Umiem przetestować, usunąć błędy semantyczne(znaczeniowe) (tzw. bugi) i sprawdzić czy program działa poprawnie**

Programowanie proceduralne(było):

- – Typy mutowalne oraz niemutowalne a przekazywanie argumentów do funkcji.
- – Zasięg zmiennych — zmienne lokalne, nielokalne oraz globalne.
- – Argumenty funkcji: pozycyjne, nazwane, domyślne, rozpakowywanie, słownikowe
- – Funkcje: „first citizen object”.
- – Funkcje: funkcje wyższego rzędu.
- – Funkcje anonimowe.
- – Dopełnienia funkcji.
- – Zastosowania funkcji.

Programowanie obiektowe (będzie):

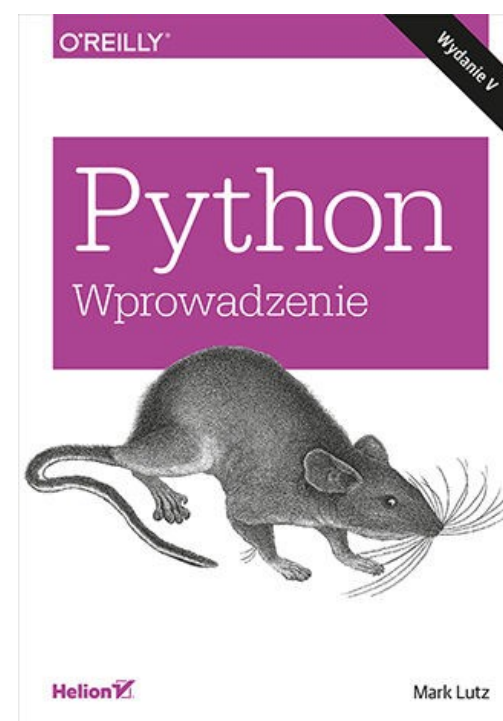
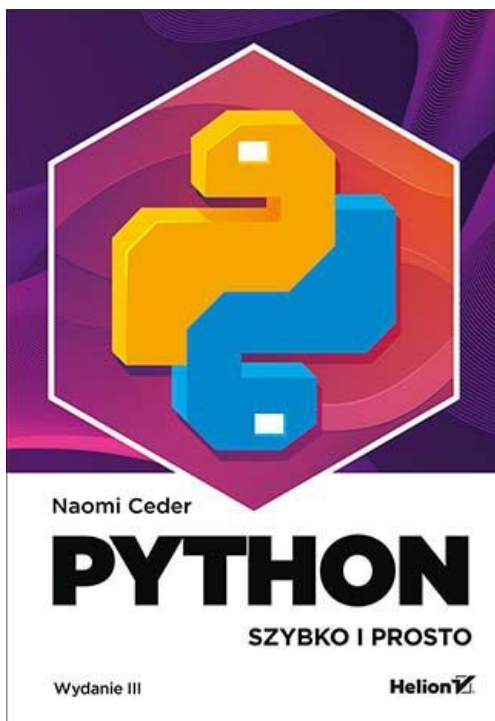
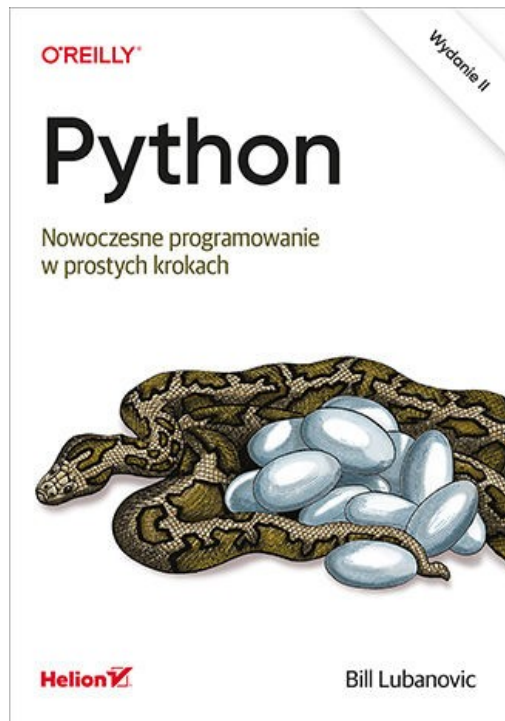
- Klasa a obiekt
- Konstruktor i destruktork.
- Metody specjalne – przeciążanie operatorów.
- Biblioteki:
 - numpy – operacje na wektorach oraz tablicach
 - matplotlib – tworzenie wykresów
 - scipy – zastosowania w fizyce
- Dziedziczenie jednokrotne klas.
- Dekoratory.
- Metody statyczne oraz pola statyczne
- Dziedziczenie wielokrotne klas.
- Metody i pola: publiczne, chronione oraz prywatne
- Podsumowanie. Filozofia programowania

Cel zajęć:

- Poznanie niektórych zaawansowanych technik programowania:
 - Metodę programowania obiektowego – zasad i filozofii
 - Poznanie niektórych podstawowych bibliotek
- Umiejętność znalezienia lub wymyślenia, zrozumienia i zastosowania (implementacji) odpowiedniego algorytmu. **CIĄGLE!**
- Umiejętność usuwania błędów składniowych oraz znaczeniowych (semantycznych) —
[https://pl.wikipedia.org/wiki/B%C5%82%C4%85d_\(informatyka\)](https://pl.wikipedia.org/wiki/B%C5%82%C4%85d_(informatyka))
CIĄGLE!



Literatura:



Inne pomoce naukowe:

- www.python.org
 - <https://www.python.org/doc/> — podręcznik języka („manual”)
 - <https://wiki.python.org/moin/>
- <https://learning-python.com/>
- <https://www.learnpython.org>
- <https://python-course.eu/python-tutorial/>
- <https://python.swaroopch.com/>
- <http://ricardoduarte.github.io/python-for-developers/>
- https://pl.wikibooks.org/wiki/Zanurkuj_w_Pythonie

Język programowania

- Składa się z jasno i precyzyjnie zdefiniowanych pojęć, reguł, symboli oraz słów kluczowych.
- Tylko program który napisany zgodnie ze składnią oraz innymi regułami jest poprawny (niestety jest to tylko warunek konieczny a nie wystarczający)
- Słowa kluczowe są zastrzeżone i nie mogą występować w innych kontekstach
- Znaczenie symboli możemy rozszerzać, modyfikować
- Ulega ewolucji (powolnym zmianom zachowując tzw. kompatybilność w górę)
- Bardzo rzadko zdarzają się rewolucje powodujące zerwanie zgodności (kompatybilności) z wcześniejszymi wersjami języka

Python — elementy programu:

- Składniki języka
- Typy danych
- Syntaktyki (składni lub gramatyki) oraz semantyki (czyli znaczenia)
 - Instrukcje: proste i złożone
 - Instrukcje warunkowe
 - Instrukcje iteracyjne
 - Funkcje
 - Klasy



Składniki języka — symbole(=59) i słowa kluczowe(=35) — Python 3.11

```
help("symbols")
```

Here is a list of the punctuation symbols which Python assigns special meaning to. Enter any symbol to get more help.

!=	+	<=	—
"	+=	<>	`
"""	,	==	b"
%	-	>	b'
%=	-=	>=	f"
&	.	>>	f'
&=	...	>>=	j
'	/	@	r"
'''	//	J	r'
(	//=	[	u"
)	/=	\	u'
*	:	]	
**	<	^	=
**=	<<	^=	~
*=	<<=	—	

```
help("keywords")
```

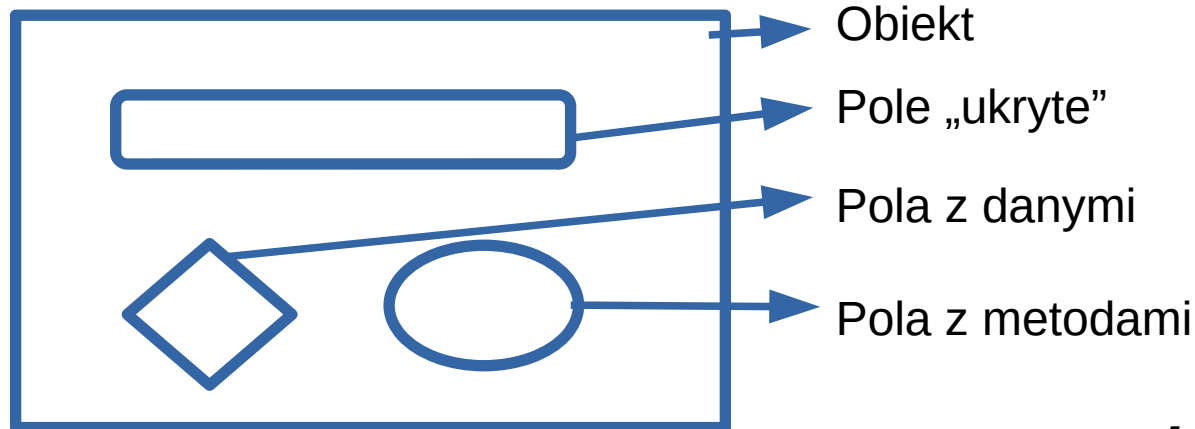
Here is a list of the Python keywords. Enter any keyword to get more help.

False	class	from	or
None	continue	global	pass
True	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield
break	for	not	

Python 3.10 oraz 3.11 mają takie same składniki języka

Obiekt — podstawy element

- W Pythonie wszystko jest obiektem. Obiekty składają się z części zwanymi polami
- Obiekt — to struktura składająca się z:
 - Pola dane — w których są informacje
 - Pola metody — w których są zapisane funkcje, pozwalające operować na obiekcie
 - Części „ukrytej” — w której zapisane są „różne” rzeczy które pozwalają aby to wszystko po prostu działało.



Obiekt czy klasa

- Klasy:
 - są zdefiniowane poprzez odpowiednie konstrukcje języka
 - Konkretna klasa definiuje jednoznacznie typ danych o takiej samej nazwie jak nazwa klasy
 - Daną klasa można tylko jeden raz zdefiniować — może być tylko jeden konkretny typ danych
- Obiekty
 - Są instancją (konkretną realizacją) klasy czyli są budowane wg przepisu zawartego w klasie
 - Są tworzone automatycznie poprzez odpowiednie konstrukcje języka
 - Obiektów w programie może być wiele, ale:
 - Każdy obiekt ma takie same metody
 - Każdy obiekt może mieć (i na ogół ma) różne wartości w polach danych

Identyfikator a zmienna

- Identyfikator:
 - Jednoznacznie identyfikuje obiekt
 - Wskazuje gdzie znajduje się obiekt
 - Najczęściej wartością identyfikatora jest adres w pamięci komputera gdzie znajduje się obiekt
- Zmienne:
 - Formalnie: są to nazwane obszary pamięci
 - W Pythonie, zmienne służą tylko do zapamiętywania identyfikatora obiektu

Wniosek: Ponieważ wszystko jest obiektem, a obiekt to zmienna — w Pythonie w pewnym sensie metody oraz dane możemy traktować podobnie, oczywiście do pewnego stopnia

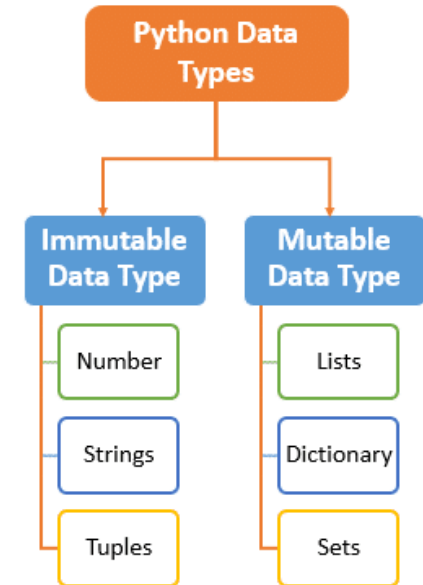
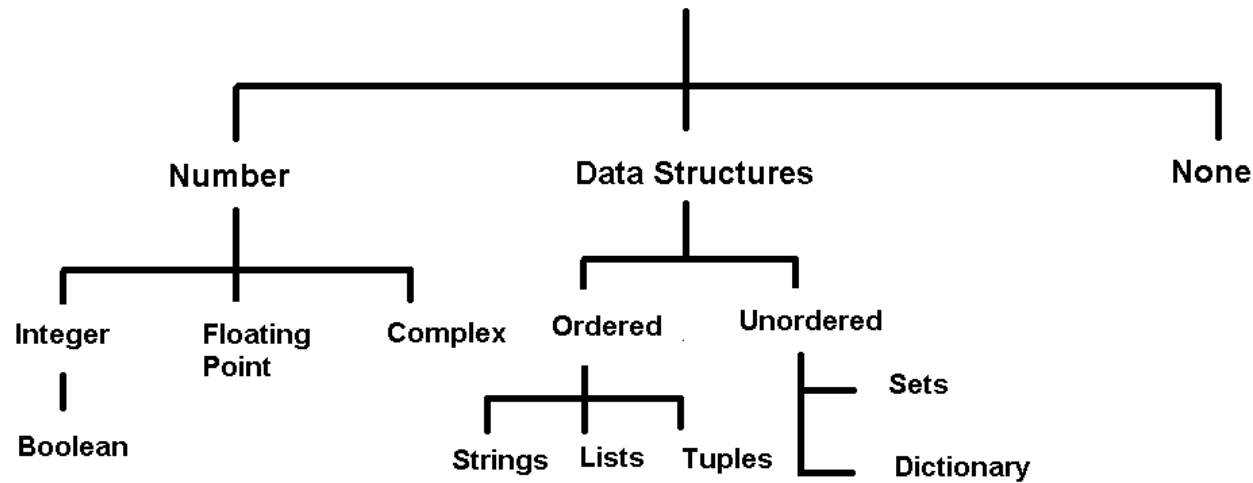
Co to oznacza w praktyce?

- Typ zmiennej — jest to typ obiektu wskazywanego przez zmienną
- „Wywołanie” zmiennej powoduje wywołanie odpowiedniej metody obiektu w zależności od kontekstu, np:
 - Jeśli chcemy wyświetlić zmienną (instrukcja `print`) niejawnie wywoływana jest metoda zwracająca tekstową formę obiektu:
`print(zmienna)`
 - Jeśli chcemy zmiennej użyć w wyrażeniach, wywoływana jest metoda zwracająca wartość obiektu
`zmienna1=zmienna2+zmienna3`

Nazwy zmiennych:

- Nazwy zbudowane są wg reguł:
 - Muszą zaczynać się od podkreślenia „_” albo małej czy dużej litery
 - Kolejnymi znakami w nazwie mogą być litery, cyfry lub znak podkreślenia
 - Mogą być dowolnej długości
 - Duże i małe litery są rozróżnialne
 - Nie mogą być tzw. słowami kluczowymi Pythona
 - Nazwy zaczynające się od znaku podkreślenia mają specjalne znaczenie
- „Same zmienne” mogą „zapamiętać” tylko identyfikator obiektu

Python Data Types



- Typy danych w Pythonie można podzielić na różne sposoby:
 - Proste i złożone
 - Mutowalne i niemutowalne
 - Kolekcje i sekwencje
 - Wbudowane i własne
 - ...

Program

- Składa się z instrukcji
 - Instrukcja prosta — najmniejszy samodzielny element języka programowania nadający się do uruchomienia przez procesor
 - Instrukcja złożona — składa się z instrukcji prostych, instrukcji grupującej oraz nagłówka
 - Komentarzy
 - Specjalnych konstrukcji z informacjami dla kompilatora czy interpretera
- Instrukcje są zapisane w pliku tekstowym(gdzie zapisujemy kod źródłowy programu) lub binarnym
- Pliki z programem można grupować np. biblioteki, moduły

Program w Pythonie

- W jednej linii powinna być tylko jedna instrukcja (można ją też zakończyć średnikiem)
- Jeśli chcemy umieścić dwie lub więcej instrukcji, wtedy instrukcje te oddzielamy średnikiem
- Komentarz zaczyna się od znaku # i obowiązuje do końca linii
- Instrukcja złożona
- Instrukcje czytane są od pierwszej linii z pliku który uruchamiamy, ale kolejność wykonywania instrukcji generalnie zależy od przebiegu programu
- Przebieg programu zmodyfikowany jest przez:
 - Wywołanie funkcji
 - Instrukcje warunkowe
 - Pętle
 - Import modułów (pozwalają nam zwiększyć funkcjonalność Pythona)



Dziękuję za uwagę