Wrocław University of Science and Technology

Wstęp do programowania

INP003203L

Semestr zimowy 2020/2021

Poniedziałek, 8:00 - 9:00

sala wirtualna
– zajęcia online

Sylwia Majchrowska

sylwia.majchrowska@pwr.edu.pl

https://majsylw.netlify.app/teaching/

pokój 213, budynek L-1

Wrocław University of Science and Technology

Plan na dziś

1. Pytania quizowe - kahoot
2. Zadanie 6 – analiza kodu – github
3. Kilka dodatkowych słów o operatorach i logice

Wrocław University of Science and Technology

Spojrzenie na kalendarz

		PAŹDZIERNIK				LISTOPAD					GRUDZIEŃ				STYCZEŃ				LUTY			
PN	28	5	12	19	26	2	9 K1	16	23	30 K2	7	14	21 K3	28	4	11	18	25 T1 Kolokwium	1 T2 Pn N	8	15	22
WT	29	6	13	20	27	3	10	17	24	1	8	15	22 Śr P	29	5	12	19	26	2	9	16	23
ŚR	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	3	10	17	24
CZ	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25
PT	2 Pt N	9 Pt P	16	23	30	6	13 Śr P	20	27	4	11	18	25	1	8	15	22	29	5	12	19	26
SO	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	6	13	20	27
N	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	31	7	14	21	28
P - PARZYSTY N - NIEPARZYSTY	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N

Poniedziałek, 7:30 - 9:00
sala wirtualna

Konsultacje:
Poniedziałki, 19:30 - 20:30
Środy, 19:30 – 20:30

Wrocław University of Science and Technology

Co będziemy dalej robić?

1. Wprowadzenie –zapoznanie ze środowiskiem – pierwszy program
2. Projektowanie programu. Objaśnianie kodu za pomocą komentarzy. Pisanie na ekran. Zmienne. Proste funkcje.
3. Interakcja z użytkownikiem. Przekazywanie argumentów do funkcji.
4. Obliczenia matematyczne. Funkcje zwracające wartość.
5. Sterowanie -konstrukcja if-else. Operatory relacji.
6. Zagnieżdżone struktury warunkowe. Operatory logiczne.
7. Sterowanie - pętle for, while.
8. Pętle zagnieżdżone.
9. Poszerzenie wiadomości o funkcjach – import modułów.

Wrocław University of Science and Technology

Operator warunkowy

w Pythonie

W języku python:

```
zmienna = instrukcja_true if warunek else instrukcja_false
```

W językach C, C++, Java, C#, Perl, PHP (od wersji 5.3) i Ruby:

```
zmienna = warunek ? instrukcja_true : instrukcja_false
```

daje to taki sam rezultat, co:

```
if warunek:
    zmienna = instrukcja_true
else:
    zmienna = instrukcja_false
```

```
a = 1
print( "różne od zera" if a != 0 else "równe zero" )
a = 0
print( "różne od zera" if a != 0 else "równe zero" )
```

https://github.com/majsylw/Introduction-to-programming-in-python/wiki/6.-Instrukcje-warunkowe#operator-warunkowy-w-pythonie

Wrocław University of Science and Technology

Operatory bitowe

Arg A	Arg B	A & B	A | B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Znaczenie	Operator
Alternatywa bitowa (or na bitach)	|
Koniunkcja bitowa (and na bitach)	&
Negacja bitowa (not na bitach)	~
Alternatywa wykluczająca (xor na bitach)	^

Arg A	~ Arg A
0	1
1	0

Trochę prościej:

- **&** wymaga dwóch 1 aby dać 1.
- **|** wymaga przynajmniej jednej 1 aby dać 1.
- **^** wymaga dokładnie jednej 1 aby dać 1.
- Wykorzystywane zmienne muszą być całkowite!
- Operujemy na każdym bicie danej oddzielnie.

```
i = 15 # dwójkowo: 00000000000000000000000000001111
j = 22 # dwójkowo: 00000000000000000000000000010110
print(i and j)   # True
print(i & j)     #  00000000000000000000000000000110
```

Wrocław University of Science and Technology

Logika trzech odpowiedzi

hierarchia

Znaczenie	Operator
Alternatywa (lub)	or
Koniunkcja (i)	and
Zaprzeczenie (nie)	not

Koniunkcja - and

<wyrażenie A>	<wyrażenie B>	A and B
True	True	True
True	False	False
False	True	False
False	False	False
True	None	None
False	None	False

None and False -> None

None and True -> None

python

Alternatywa - or

<wyrażenie A>	<wyrażenie B>	A or B
True	True	True
True	False	True
False	True	True
False	False	False
True	None	True
False	None	None

None or False -> False

None or True -> True

python

Negacja - not

<wyrażenie>	not <wyrażenie>
True	False
False	True
None	None (w pythonie True)

Wrocław University of Science and Technology

False ?

```
i = "To była prawda!"
if None:
    print(i)
if []:
    print(i)
if [1, 2]:
    print(i)
if ():
    print(i)
if (False, False):
    print(i)
if "":
    print(i)
if " ":
    print(i)
if "Ala":
    print(i)
print("2" == True)
```

To jest prawda!

To jest prawda!

To jest prawda!

To jest prawda!

Uwaga! Bezpośrednie porównanie zmiennych różnego typu da False!