

# Wstęp do programowania

INP003203L

rok akademicki 2020/21

semestr zimowy

## Laboratorium 1

Karol Tarnowski

[karol.tarnowski@pwr.edu.pl](mailto:karol.tarnowski@pwr.edu.pl)

L-1 p. 220



# Plan prezentacji (1)

- Wprowadzenie
  - Dlaczego warto uczyć się programować?
  - Sprzęt i oprogramowanie
  - Jak komputer przechowuje dane?
  - Jak działa procesor?
  - Co to jest język wysokiego poziomu?
  - Czym się różni interpreter od kompilatora?
  - Co jest potrzebne do pisania programów w języku python?



# Plan prezentacji (2)

- Interpreter pythona
  - Tryb interaktywny
  - Tryb skryptowy
- Proces programowania

# Dlaczego warto uczyć się programowania?

- Programowanie uczy analitycznego myślenia
- Programowanie rozwija kreatywność, pracy w grupie
- Programowanie to umiejętność, na którą rośnie zapotrzebowanie



# Sprzęt i oprogramowanie

- Sprzęt (hardware) to wszystkie urządzenia, z których zbudowany jest komputer
- Oprogramowanie (software) steruje pracą komputera
- Komputer jest elastycznym narzędziem, dzięki oprogramowaniu



# Sprzęt i oprogramowanie

- Sprzęt:
  - procesor
  - pamięć operacyjna
  - nośniki danych
  - urządzenia wejściowe
  - urządzenia wyjściowe

# W jaki sposób komputer przechowuje dane?

- Pamięć komputera możemy wyobrażać sobie jako zestaw przełączników, które mogą przyjmować jeden z dwóch stanów (podobnie jak przełącznik oświetlenia)
- Pojedynczy przełącznik nazywamy bitem (binary digit)
- Grupę ośmiu bitów nazywamy bajtem

OFF	ON	OFF	OFF	ON	ON	OFF	ON
0	1	0	0	1	1	0	1

# W jaki sposób komputer przechowuje dane?

- Ciąg zer i jedynek można interpretować jako liczbę całkowitą zapisaną w systemie dwójkowym
- Podobnie liczby można interpretować jako znaki (litery, cyfry, znaki specjalne)
- Stosując bardziej zaawansowane kodowania można w ten sposób zapisywać także liczby rzeczywiste





# Jak działa procesor?

- Procesor wykonuje bardzo proste operacje („działania na przełącznikach”), np. odczytaj dane, dodaj dwie liczby, ...
- Lista instrukcji procesora także jest zakodowana w postaci liczb
- Program komputerowy to lista instrukcji zapisanych w języku maszynowym

# Co to jest język wysokiego poziomu?

- Aby programowanie umożliwić programowanie bez znajomości instrukcji procesora stworzono języki wysokiego poziomu (języki programowania)
- Wybrane języki programowania:
  - Fortran
  - Pascal
  - C/C++
  - Java
  - Python

# Czym się różni kompilator od interpretera?

1.

Kompilator tłumaczy program w języku wysokiego poziomu na program w języku maszynowym

```
print('Hello')  
print('world!')  
...
```

kompilator

```
10100001  
10111000  
...
```

2.

Program w języku maszynowym jest wykonywany na procesorze

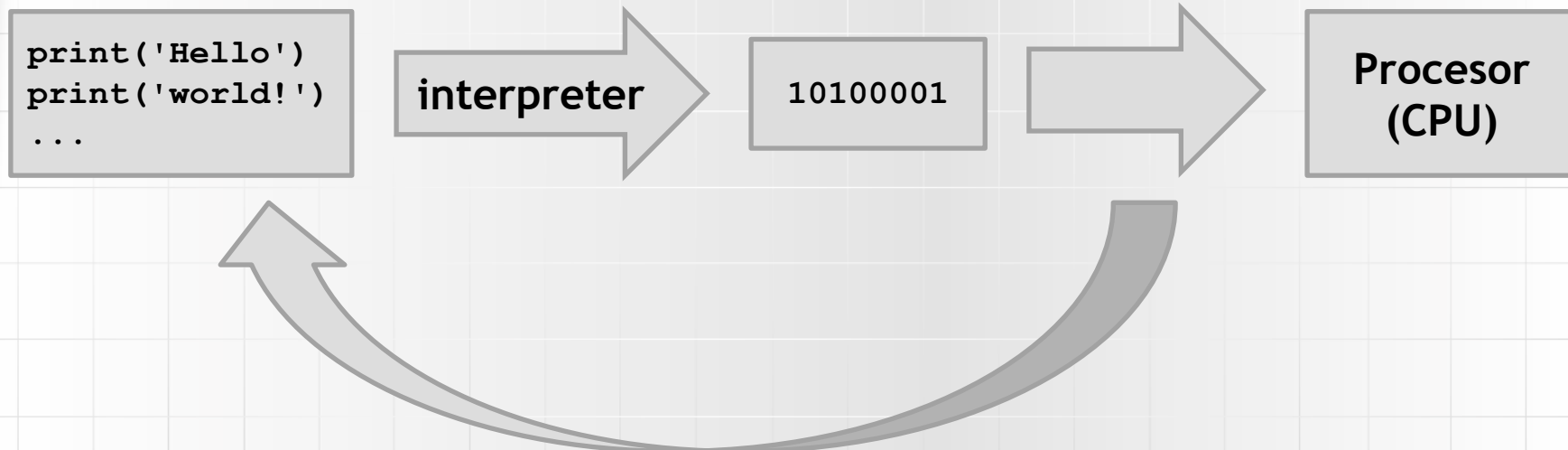
```
10100001  
10111000  
...
```

Procesor (CPU)

# Czym się różni kompilator od interpretera?

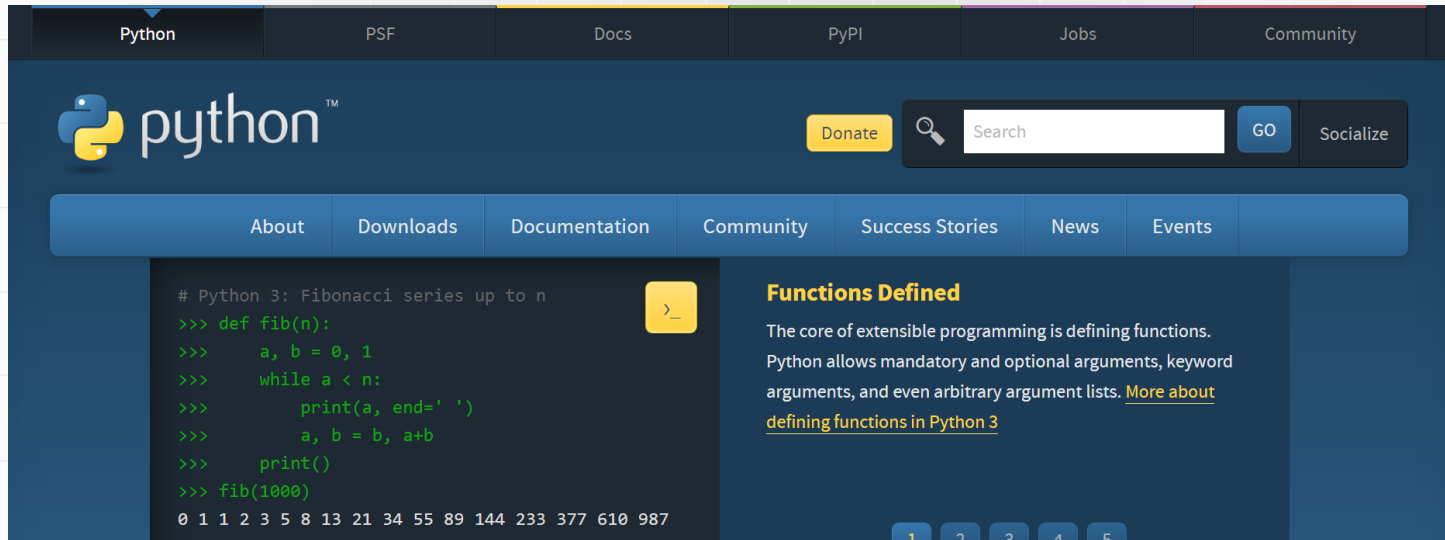
Interpreter tłumaczy instrukcję języka wysokiego poziomu na instrukcje języka maszynowego, która jest wykonywana

Następnie przechodzi do kolejnej instrukcji w języku wysokiego poziomu



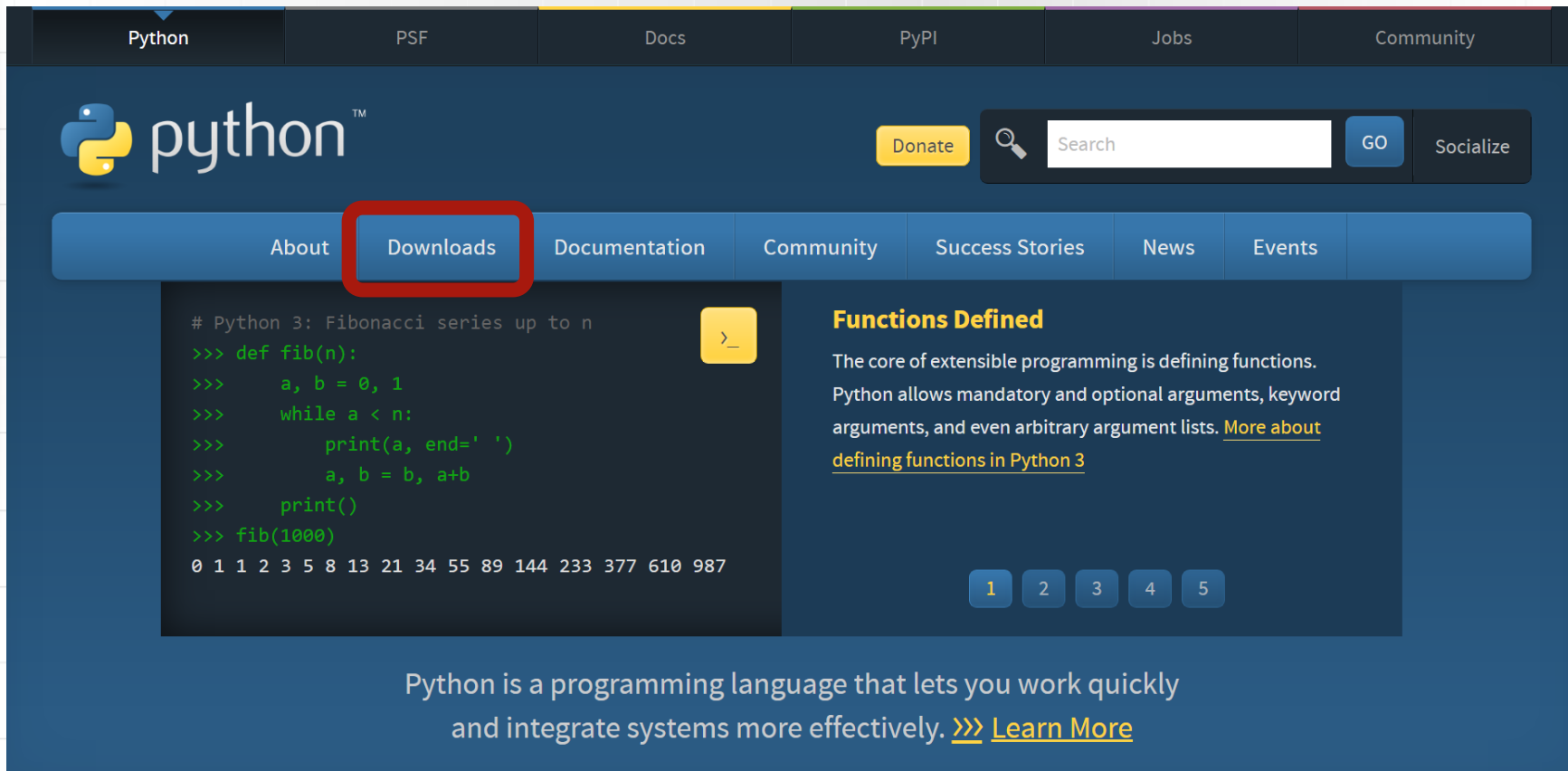
# Co jest potrzebne do pisania programów w języku python?

- Interpreter języka python ([www.python.org](http://www.python.org))



- Zintegrowane środowisko programistyczne (*Integrated DeveLopment Environment*)

# Co jest potrzebne do pisania programów w języku python?



The screenshot shows the Python.org website. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. The main navigation bar has links for About, Downloads (highlighted with a red box), Documentation, Community, Success Stories, News, and Events. The main content area features a code editor with a Python 3 Fibonacci function and its output, a 'Functions Defined' section, and a footer with the text 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

**Functions Defined**

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

# Co jest potrzebne do pisania programów w języku python?



The screenshot shows the Python.org website. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. The main content area features the Python logo, a search bar, and a 'Download Python 3.9.0' button highlighted with a red box. Below the button, there are links for 'Python for Windows, Linux/UNIX, Mac OS X, Other', 'Prereleases', and 'Docker images'. A large orange arrow points from the highlighted button down to the 'Active Python Releases' section.

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

**Download the latest version for Windows**

**Download Python 3.9.0**

Want to help test development versions of Python? [Prereleases](#),  
[Docker images](#)

Looking for Python 2.7? See below for specific releases



Active Python Releases



# Co jest potrzebne do pisania programów w języku python?

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more	
<b>Python 3.9.0</b>	Oct. 5, 2020	 Download	<a href="#">Release Notes</a>
<b>Python 3.8.6</b>	Sept. 24, 2020	 Download	<a href="#">Release Notes</a>
<b>Python 3.5.10</b>	Sept. 5, 2020	 Download	<a href="#">Release Notes</a>
<b>Python 3.7.9</b>	Aug. 17, 2020	 Download	<a href="#">Release Notes</a>
<b>Python 3.6.12</b>	Aug. 17, 2020	 Download	<a href="#">Release Notes</a>
<b>Python 3.8.5</b>	July 20, 2020	 Download	<a href="#">Release Notes</a>
<b>Python 3.8.4</b>	July 13, 2020	 Download	<a href="#">Release Notes</a>

[View older releases](#)

## Licenses

All Python releases are [Open Source](#).  
Historically, most, but not all,

## Sources

For most Unix systems, you must  
download and compile the source

## Alternative Implementations

This site hosts the "traditional"

## History

Python was created in the early  
1990s by Guido van Rossum at



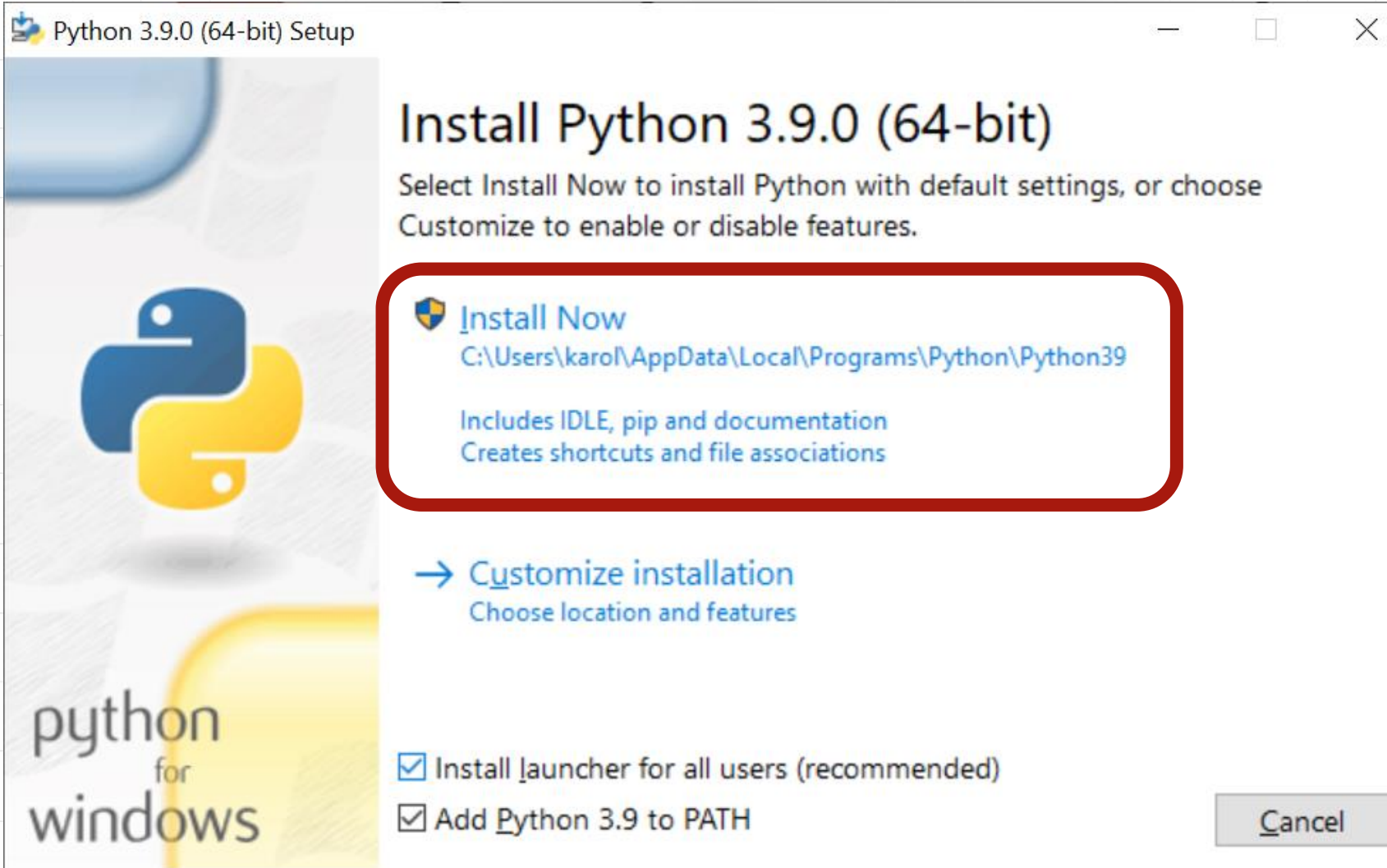


# Co jest potrzebne do pisania programów w języku python?

## Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		e19e75ec81dd04de27797bf3f9d918fd	26724009	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		6ebfe157f6e88d9eabfbaf3fa92129f6	18866140	<a href="#">SIG</a>
<a href="#">macOS 64-bit installer</a>	Mac OS X	for OS X 10.9 and later	16ca86fa3467e75bade26b8a9703c27f	31132316	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		9ea6fc676f0fa3b95af3c5b3400120d6	8757017	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	60d0d94337ef657c2cca1d3d9a6dd94b	8387074	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	b61a33dc28f13b561452f3089c87eb63	28158664	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	733df85afb160482c5636ca09b89c4c8	1364352	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		d81fc534080e10bb4172ad7ae3da5247	7553872	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		4a2812db8ab9f2e522c96c7728cfcccb	27066912	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		cdbfa799e6760c13d06d0c2374110aa3	1327384	<a href="#">SIG</a>

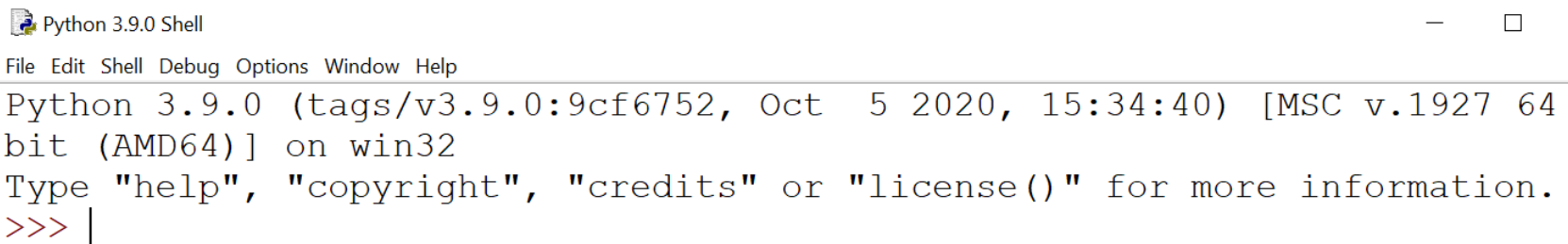
# Co jest potrzebne do pisania programów w języku python?





# Interpreter pythona

- Interpreter Pythona może pracować w trybie interaktywnym oraz skryptowym
- Interpreter działający w trybie interaktywnym nazywany jest powłoką Pythona (*python shell*)



```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64
bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```



# Interpreter pythona

- W trybie interaktywnym interpreter oczekuje na polecenia pythona wpisywane z klawiatury
- Przykładowo

```
*Python 3.9.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello world!')
```



# Interpreter pythona

- W trybie interaktywnym interpreter oczekuje na polecenia pythona wpisywane z klawiatury
- Przykładowo

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello world!')
Hello world!
>>> |
```



# Interpreter pythona

Python 3.9.0 Shell



File Edit Shell Debug Options Window Help

Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

```
>>> print('Hello world!')
```

```
Hello world!
```

```
>>> print('Litwo! Ojczyzno moja! Ty jesteś jak zdrowie.')
```

```
Litwo! Ojczyzno moja! Ty jesteś jak zdrowie.
```

```
>>> |
```



# Interpreter pythona

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello world!')
Hello world!
>>> print('Litwo! Ojczyzna moja! Ty jesteś jak zdrowie.')
Litwo! Ojczyzna moja! Ty jesteś jak zdrowie.
>>> print('Ile Cię trzeba cenić, ten tylko się dowie,')
Ile Cię trzeba cenić, ten tylko się dowie,
>>> |
```



# Interpreter pythona

- Zaletą trybu interaktywnego jest natychmiastowe wyświetlenie informacji o błędzie, jeśli jakiś popełnimy





# Interpreter pythona

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello world!')
Hello world!
>>> print('Litwo! Ojczyzno moja! Ty jesteś jak zdrowie.')
Litwo! Ojczyzno moja! Ty jesteś jak zdrowie.
>>> print('Ile Cię trzeba cenić, ten tylko się dowie,')
Ile Cię trzeba cenić, ten tylko się dowie,
>>> print('Kto Cię stracił. Dziś piękność Twą w całej ozdobie)
SyntaxError: EOL while scanning string literal
>>>
```



# Interpreter pythona

- Tryb interaktywny jest użyteczny podczas testowania kodu, ale wpisywane polecenia nie będą zapisane jako program
- Kod programu przechowywany jest w pliku tekstowym
- Prosty sposób pracy ze skryptami zapewnia zintegrowane środowisko programistyczne (Integrated Development Environment)



# Interpreter pythona

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
New File Ctrl+N
Open... Ctrl+O
Open Module... Alt+M
Recent Files
Module Browser Alt+C
Path Browser
Save Ctrl+S
Save As... Ctrl+Shift+S
Save Copy As... Alt+Shift+S
Print Window Ctrl+P
Close Alt+F4
Exit Ctrl+Q
ags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [
it (AMD64)] on win32
copyright", "credits" or "license()" for more
o world!')
! Ojczyzno moja! Ty jesteś jak zdrowie.')
moja! Ty jesteś jak zdrowie.
Cię trzeba cenić, ten tylko się dowie,')
cenić, ten tylko się dowie,
Cię stracił. Dziś piękność Twą w całej ozdob
ie)
SyntaxError: EOL while scanning string literal
>>> |
```



# Interpreter pythona

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020,
15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license(
)" for more information.
>>> print('Hello world!')
Hello world!
>>> print('Litwo! Ojczyzno moja! Ty jesteś jak z
drowie.')
```

```
*untitled*
File Edit Format Run Options Window Help
1 print('Natenczas Wojski chwycił na taśmie przypięty')
2 print('Swój róg bawoli, długi, cętkowany, kręty')
3 print('Jak wąż boa, oburącz do ust go przycisnął,')
4 print('Wzdał policzki jak banię, w oczach krwią zabłysnął,')
5 print('Zasunął w pół powieki, wciągnął w głąb pół brzucha')
6 print('I do płuc wysłał z niego cały zapas ducha.')
```



# Interpreter pythona

The screenshot shows a window titled '\*untitled\*' with a menu bar containing 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The 'File' menu is open, displaying the following options:

- New File Ctrl+N
- Open... Ctrl+O
- Open Module... Alt+M
- Recent Files
- Module Browser Alt+C
- Path Browser
- Save Ctrl+S
- Save As... Ctrl+Shift+S**
- Save Copy As... Alt+Shift+S
- Print Window Ctrl+P
- Close Alt+F4
- Exit Ctrl+Q

The main text area contains the following Python code snippet:

```
as Wojski chwycił na taśmie przypięty')  
g bawoli, długi, cętkowany, kręty')  
boa, oburącz do ust go przycisnął,')  
oliczki jak banię, w oczach krwią zabłysnął,')  
wśród powieki, wciągnął w głąb pól brzucha')  
uc wysłał z niego cały zapas ducha.')
```



# Interpreter pythona

The screenshot shows a Python IDE window titled "natenczas.py - D:/dydaktyka/202021/wdp/natenczas.py (3.9.0)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The "Run" menu is open, showing options: "Run Module" (F5), "Run... Customized" (Shift+F5), "Check Module" (Alt+X), and "Python Shell". The code in the editor consists of six print statements:

```
1 print ('...ki chwycił na taśmie przypięty')
2 print ('... , długi, cętkowany, kręty')
3 print ('...buracz do ust go przycisnął,')
4 print ('wzdał policzki jak banię, w oczach krwią zabłysnął,')
5 print ('Zasunął w pół powieki, wciągnął w głąb pół brzucha')
6 print ('I do płuc wysłał z niego cały zapas ducha.')
7 |
```



# Interpreter pythona

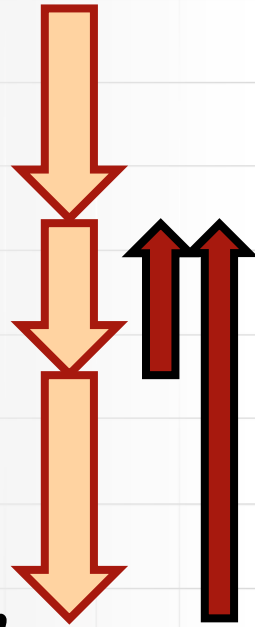
```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Type "help", "copyright", "credits" or "license(
)" for more information.
>>> print('Hello world!')
Hello world!
>>> print('Litwo! Ojczyzno moja! Ty jesteś jak z
drowie.')
Litwo! Ojczyzno moja! Ty jesteś jak zdrowie.
>>> print('Ile Cię trzeba cenić, ten tylko się d
owie,')
Ile Cię trzeba cenić, ten tylko się dowie,
>>> print('Kto Cię stracił. Dziś piękność Twą w
całej ozdobie)
SyntaxError: EOL while scanning string literal
>>>
= RESTART: D:/dydaktyka/202021/wdp/natenczas.py
Natenczas Wojski chwycił na taśmie przypięty
Swój róg bawoli, długi, cętkowany, kręty
Jak wąż boa, oburącz do ust go przycisnął,
Wzdął policzki jak banię, w oczach krwią zabłysn
ął,
Zasunął wpół powieki, wciągnął w głąb pół brzuch
a
I do płuc wysłał z niego cały zapas ducha.
>>> |

natenczas.py - D:/dydaktyka/202021/wdp/natenczas.py (3.9.0)
File Edit Format Run Options Window Help
1 print('Natenczas Wojski chwycił na taśmie przyp.
2 print('Swój róg bawoli, długi, cętkowany, kręty
3 print('Jak wąż boa, oburącz do ust go przycisną
4 print('Wzdął policzki jak banię, w oczach krwią
5 print('Zasunął wpół powieki, wciągnął w głąb pó
6 print('I do płuc wysłał z niego cały zapas duch.
7
```



# Proces programowania

1. Dokładne określenie planowanej funkcjonalności programu
2. Napisanie kodu programu
3. Sprawdzenie, czy program nie zawiera błędów składni, i ewentualnie ich naprawienie
4. Uruchomienie programu i sprawdzenie, czy działa prawidłowo (zgodnie z oczekiwaniami), ewentualna poprawa







# Absolutne minimum (1)

- zainstaluj interpreter języka Python na swoim komputerze
- uruchom interpreter w trybie interaktywnym
- wywołaj funkcję print w trybie interaktywnym



# Absolutne minimum (2)

- w zintegrowanym środowisku programistycznym przygotuj skrypt pythona, który spowoduje wypisanie na ekran wybranego cytatu (wiersza, piosenki...)