

Podstawy programowania w języku Python

Dzień 8

Dzień 8

- wyjątki
- refaktoryzacja
- Debugging
- **livecoding**
- **smtplib, BeautifulSoup, pillow**

Syntax Error = błąd w składni polecenia

Exception, Error = wyjątki – to inaczej błędy powstałe w trakcie wykonywania programu np.

- *błąd dzielenia przez zero,*
- *brak zdefiniowanej zmiennej,*
- *odwołanie się do nie istniejącego indeksu*

<https://docs.python.org/3/library/exceptions.html>

try:

raise RuntimeError("Oops!")

#raise DziwnyWyjatek

#pass

except RuntimeError:

print("Wystąpił wyjątek Runtime Error")

except:

print("Wystąpił nieokreślony wyjątek")

else:

print("Wykonam się tylko jeśli nie było żadnego wyjątku")

finally:

print("Zawsze się wykonam!")

*Staramy się wyłapywać konkretne typy wyjątków, np. **ValueError**, **NameError** zamiast ogólnego wyjątku **Exception**.*

Bloki except deklarujemy od szczegółu do ogółu

*W przypadku kilku błędów, **tylko** jeden blok except zostanie wywołany, ten najwyżej.*

Wyjątek można samemu wywołać celowo!

raise ValueError(„Nasz komunikat”)

raise DziwnyWyjatek

Proces poprawiania struktury kodu,
bez zmiany jego funkcjonalności.

Odnajdywanie i usuwanie błędów z programu.

Bug – z ang. robak

1947 – Admiral Grace Hopper

(Harvard Mark I)


9/9

0800 Anttan started
1000 " stopped - anttan ✓

1300 (032) HP-MC { 1.2700 9.037 847 023
2.130476415 (032) 4.6159250
(033) PRO 2 2.130476415
convd 2.130676415

Relays 6-2 in 033 failed special speed test
in relay 11.000 test.

Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

1630 Anttan started.
1700 closed down.



- 1) debugowanie za pomocą **print()**
- 2) debugowanie za pomocą **debuggera**
 - program / moduł ułatwiający debuggowanie
 - zatrzymanie programu w dowolnym miejscu
 - wykonywanie instrukcji krok po kroku
 - podgląd aktualnego stanu zmiennych, pamięci
- 3) debugowanie za pomocą **kaczki?**

Debugger

```
1  # -*- coding: utf-8 -*-
2
3  dni_tygodnia = ['poniedziałek', 'wtorek', 'środa', 'czwartek', 'piątek', 'sobota', 'niedziela'] dni_tygodnia: <class 'list': ['pon
4  weekend = ['sobota', 'niedziela'] weekend:
5
6  for dzien in dni_tygodnia: dzien: 'sobota'
7  if dzien in weekend:
8      print('Jest {} więc odpoczywamy'.format(dzien))
9  else:
10     print('Jest {} więc pracujemy'.format(dzien))
```

Debug: day7 x

Debugger Console →

Frames → Variables

MainThread

<module>, day7.py:7

execfile, _pydev_execfile.py:18

run, pydevd.py:1068

main, pydevd.py:1658

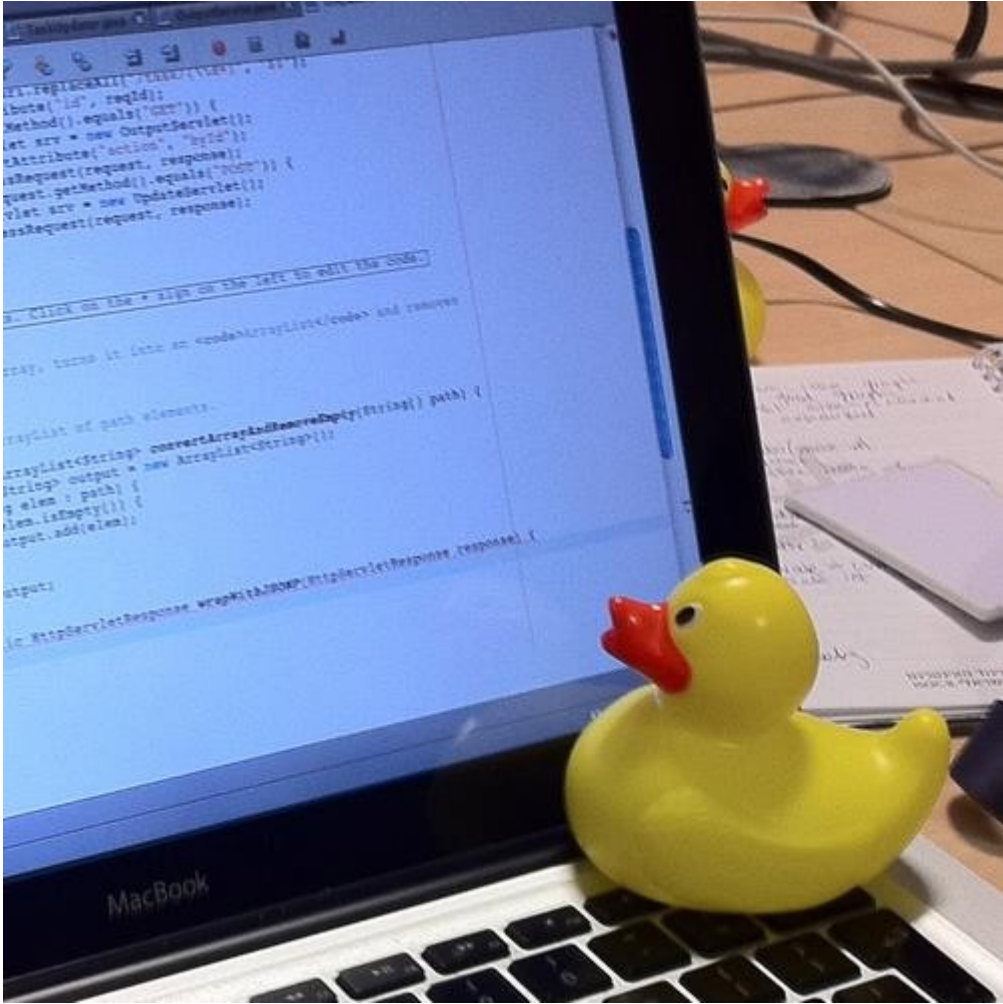
<module>, pydevd.py:1664

Special Variables

dni_tygodnia = {list} <class 'list': ['poniedziałek', 'wtorek', 'środa', 'czwartek', 'piątek', 'sobota', 'niedziela']

dzien = {str} 'sobota'

weekend = {list} <class 'list': ['sobota', 'niedziela']



Przydatne moduły:

- **smtplib** – Simple Mail Transfer Protocole
- **imaplib** – obsługa poczty IMAP
- **email.mime.MimeText** – format przesyłania informacji MIME

<https://docs.python.org/3/library/smtplib.html#module-smtplib>

<https://docs.python.org/3.1/library/email-examples.html>

Serwer pocztowy

login: **isapy@o2.pl**

hasło: **isapython**

smtp: **poczta.o2.pl**

port: **465**

zabezpieczenie: **SSL**

Algorytm :)

- 1) importuje biblioteki
- 2) mam temat wiadomości i treść wiadomości
- 3) tworze mailera
- 4) witam się z serwerem smtp – tworzę połączenie
- 5) włączam szyfrowanie (bo chce przesłać do serwera login i hasło)
- 6) loguję się (podając login i hasło)
- 7) wysyłam maila
- 8) kończę połączenie z serwerem

Moduł do pracy ze zdjęciami

```
pip install Pillow
```

```
from PIL import Image
```

```
foto = Image.open(plik)
```

Niektóre z metod:

`copy()`, `crop()`, `filter()`, `paste()`, `resize()`, `rotate()`, `save()`

<https://pillow.readthedocs.io/en/4.0.x/>
<https://pillow.readthedocs.io/en/4.0.x/reference/Image.html>

BeautifulSoup

Moduł do pracy z plikami HTML, XML

```
from bs4 import BeautifulSoup
```

```
soup = BeautifulSoup(html_doc, 'html.parser')
```

Niektóre z metod:

get(), find(), find_all()

Koniec

DZIĘKUJĘ