

**Języki skryptowe
Python**

Uniwersytet Technologiczno-Przyrodniczy im. J.J. Śniadeckich w Bydgoszczy			
Instrukcja do ćwiczeń laboratoryjnych			
Przedmiot	Nr	Temat	Studia
Języki skryptowe	3	Wyrażenia regularne, http	SP-3

Ćwiczenie 1. Wyrażenia regularne

Funkcje obsługujące wyrażenia regularne w Pythonie znajdują się w module re.

```
>>> import re
```

1.1. Funkcja match. Funkcja `re.match` próbuje odszukać dany wzorzec na początku ciągu; jeśli go znajdzie, zwraca znaleziony obiekt `match`, jako powodzenie (`Match`) lub niepowodzenie (`None`).

Składania dla tej funkcji to:

```
re.match(pattern, string, flags=0)
```

`pattern` - wyrażenie regularne wykorzystywane do dopasowania
`string` - ciąg znaków, które będą przeszukiwane aby dopasować wzorzec do początku łańcucha
`flags` - modyfikatory; wykorzystywane w procesie wyszukiwania

Jak to wygląda w praktyce?

Example 1

```
r = re.match("P.t", "Python")
print(r.group())
```

1.2. Funkcja search. Funkcja `re.search` wyszukuje pierwsze dopasowanie wzoru do ciągu znaków. Funkcja `re.search` zwraca obiekt `match`, jako powodzenie (`Match`) lub niepowodzenie (`None`).

Składania dla tej funkcji to:

```
re.search(pattern, string, flags=0)
```

Jak to wygląda w praktyce?

Example 1

```
value="voorheesville"
m=re.search("(vi.*)", value)
if m:
    print("search:", m.group(1))
m=re.match("(vi.*)", value)
if m:
    print("match:", m.group(1))
```

Example 2

```
x = re.search("cat","A cat and a rat can't be friends.")
print(x)
x = re.search("cow","A cat and a rat can't be friends.")
print(x)
```

Example 3

```
s = 'BEGIN hello world END'
mo = re.search('BEGIN (.*) END', s)
print(mo.group(1))
```

1.3. Funkcja compile. Funkcja `re.compile` kompiluje wyrażenie regularne i zwraca odpowiedni obiekt.

Składania dla tej funkcji to:

```
re.compile(pattern, flags=0)
```

Jak to wygląda w praktyce?

Example 1

```
patt = re.compile("\w+")
```

Opracowanie: mgr inż. Sandra Śmigiel
ZTC WTIE

Języki skryptowe Python

```
words = patt.findall("Hello World")
print(words)
```

Example 2

```
s = 'Hello world, this is a test!'
print(re.findall(r'\S+', s))
print(re.sub( r'\S+', 'WORD', s))
```

1.4. Funkcja findall. Funkcja `re.findall` zwraca wszystkie (nie nachodzące na siebie) wystąpienia wzorca w danym łańcuchu.

Składania dla tej funkcji to:

```
findall(pattern, string, flags=0)
```

Jak to wygląda w praktyce?

Example 1

```
sample="Weronika ma 15 lat, a Daniel ma 27 lat. Edward jest o 97 lat od      niego starszy i jego dziadek
Oskar ma 102 lata"
ages = re.findall(r'\d{1,3}',sample)
names = re.findall(r'[A-Z][a-z]*',sample)
print(ages)
print(names)
```

Example 2

```
value="abc 123 def 456 dot map pat"
list=re.findall("[dp]\w+", value)
print(list)
```

1.5. Funkcja sub. Funkcja `re.sub` to funkcja podmiany, która zamienia wszystkie wystąpienia wzorca w napisie na podany ciąg.

Składania dla tej funkcji to:

```
re.sub(pattern, repl, string, max=0)
```

repl - ciąg funkcji/znaków

Jak to wygląda w praktyce?

Example 1

```
str = "yes I said yes I will Yes."
res = re.sub("[yY]es", "no", str)
print (res)
```

Example 2

```
v = "running eating reading"
v = re.sub(r"r.*?ing", "ring", v)
print(v)
```

1.6. Funkcja split. Funkcja `re.split` dzieli napis wg podanego wzorca.

Składania dla tej funkcji to:

```
re.split(pattern, string, max=0)
```

Jak to wygląda w praktyce?

Example 1

```
value="one 1 two 2 three 3"
result=re.split("\D+", value)
for element in result:
    print (element)
```

Example 2

Języki skryptowe Python

```
a = "Potrzeba było dwóch uderzeń aby wydostać się z bunkra."
sample = a.split(" ",3)
print(sample)
```

1.7. Podstawowe wzorce wyrażeń regularnych

Identifiers:	Modifiers:	White Space Charts:
\d = any number \D = anything but a number \s = space \S = anything but a space \w = any letter \W = anything but a letter . = any character, except for a new line \b = space around whole words \. = period. must use backslash, because . normally means any character.	{1,3} = for digits, u expect 1-3 counts of digits, or "places" + = match 1 or more ? = match 0 or 1 repetitions. * = match 0 or MORE repetitions \$ = matches at the end of string ^ = matches start of a string = matches either/or. Example x y = will match either x or y [] = range, or "variance" {x} = expect to see this amount of the preceding code. {x,y} = expect to see this x-y amounts of the precedng code	\n = new line \s = space \t = tab \e = escape \f = form feed \r = carriage return
Chars: . + * ? [] \$ ^ () { } \		

Zadanie 1. Przy wykorzystaniu funkcji `re.search` dokonaj usunięcia z poniższego adresu e-mail stwierdzenia 'remove_this': jan@onremove_thiset.pl. Efektem końcowym powinien być adres jan@onet.pl

Zadanie 2. Przy wykorzystaniu funkcji `re.search` dokonaj wyznaczenia z ciągu znaków dwóch zawartych w niej adresów e-mail. Efektem końcowym powinny być adresy kamil@google.com i tomek@o2.pl

Ciąg znaków, jaki powinienes wykorzystać w zadaniu to: 'Kamil kamil@google.com, Tomek tomek@o2.pl'

Zadanie 3. Przy wykorzystaniu funkcji `re.sub` i podanego numeru telefonu (345-03-02) usuń style z numeru (efekt końcowy: 3450302)

Zadanie 4. Przy wykorzystaniu funkcji `re.compile` i podanego numeru telefonu (500-800-4623) wyróżnij poszczególne części numeru (efekt końcowy: '500','800','4623')

Zadanie 5. Przy wykorzystaniu funkcji `re.sub` w dowolnie napisanym przez Ciebie zdaniu podmień jego część na '*'.

Zadanie 6 Przy wykorzystaniu funkcji `re.findall` wyznacz z zdania „Nie lubię w poniedziałki wcześniej wstawać”:

- string rozpoczynający się od 'N'
- wyrazy 3-literowe
- pierwsze litery kolejnych wyrazów

Zadanie problemowe. Zdefiniuj listę z datami o różnym formacie (np. 15 styczeń 2015, 15.01.2015, 15.1.15). Określ wzorzec i na jego podstawie sprawdź, która z wymienionych przez Ciebie dat pasuje do wzorca.

Ćwiczenie 2. Jak pobrać zasoby z Internetu przy użyciu `urllib Package`?

Najprostszy sposób wykorzystania `urllib.request`:

```
urllib.request.urlopen(url[, data][, timeout])
```

1.1. Odczyt kodu źródłowego ze strony internetowej.

Example 1

```
print(urllib.request.urlopen("http://wp.pl").read().decode())
```

Opracowanie: mgr inż. Sandra Śmigiel
ZTC WTIE

```
print(x.read())
```

1.2. HTTP. urllib – odzwierciedla proces zgłoszenia żądania przez klienta i odpowiedzi nadesłanej przez serwer. Wykonuje to przy pomocy obiektu request, który reprezentuje zgłaszane przez HTTP żądanie.

Example 1

```
x = urllib.request.urlopen('https://www.google.com/')
print(x.read())
```

Example 2

```
URL2 = "http://pogoda.interia.pl/"
```

```
def doRequest(url):
```

```
#funkcja pomocnicza, która łączy się z zadaniem adresem URL i zwraca zawartość strony
    return urllib.request.urlopen(url).read().decode()
```

```
def example1():
```

```
    print(doRequest(URL2))
```

Example 2

```
with urllib.request.urlopen('http://www.python.org/') as f:
    print(f.read(300)) # zaprezentowanie pierwszy 300 bitów
```

1.3. Pobieranie wybranych danych ze strony www

Example 1

```
URL2 = "http://pogoda.interia.pl/"
```

```
def doRequest(url):
```

```
    return urllib.request.urlopen(url).read().decode()
```

```
def example1():
```

```
    response = doRequest(URL2)
    # http://unicode-table.com/en/#basic-latin
    pattern = re.compile('-?\d+\xb0C')
    # inny zapis
    pattern = re.compile(r'-?\d+°C')
    # sprawdzenie:
    print( pattern.findall("temperatura na dziś: -34°C") )
    # poszukiwanie w wynikach zapytania http:
    print( pattern.findall(response) )
```

Zadanie 1. Python dysponuje modulem urllib.parse w oparciu o który możliwe jest przedstawienie elementów URL – schemat adresowania, lokalizacja sieciowa, ścieżka, itp. Dokonaj na wybranej ścieżce wyznaczenie następujących zasobów:

- a) schemat adresowania
- b) port
- c) ścieżka

Zadanie problemowe. Pobieranie danych ze strony www. W oparciu o umiejętności zdobyte we wcześniejszych zadaniach z wybranej przez Ciebie strony pobierz aktualny kurs walut.