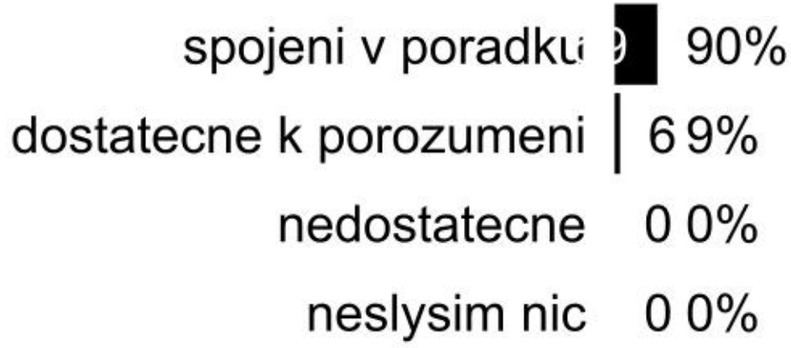
Python, základní kameny až skály I

Tomáš Svoboda
B4B33RPH, 2020-10-06
ČVUT, FEL, Katedra kybernetiky

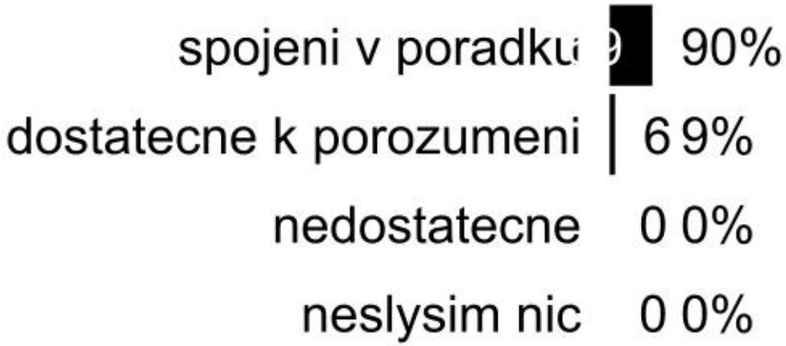


Python, základní kameny až skály I

Tomáš Svoboda

B4B33RPH, 2020-10-06

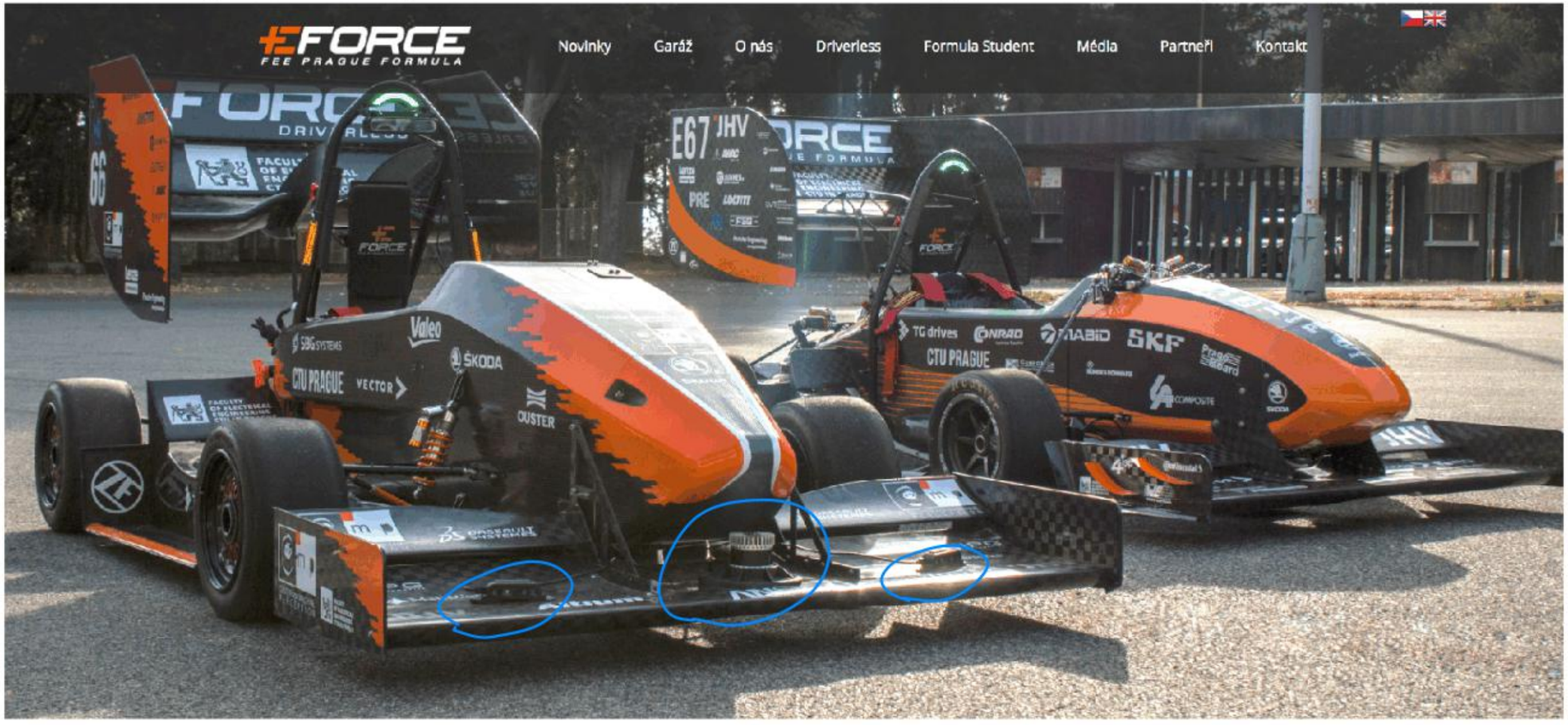
ČVUT, FEL, Katedra kybernetiky

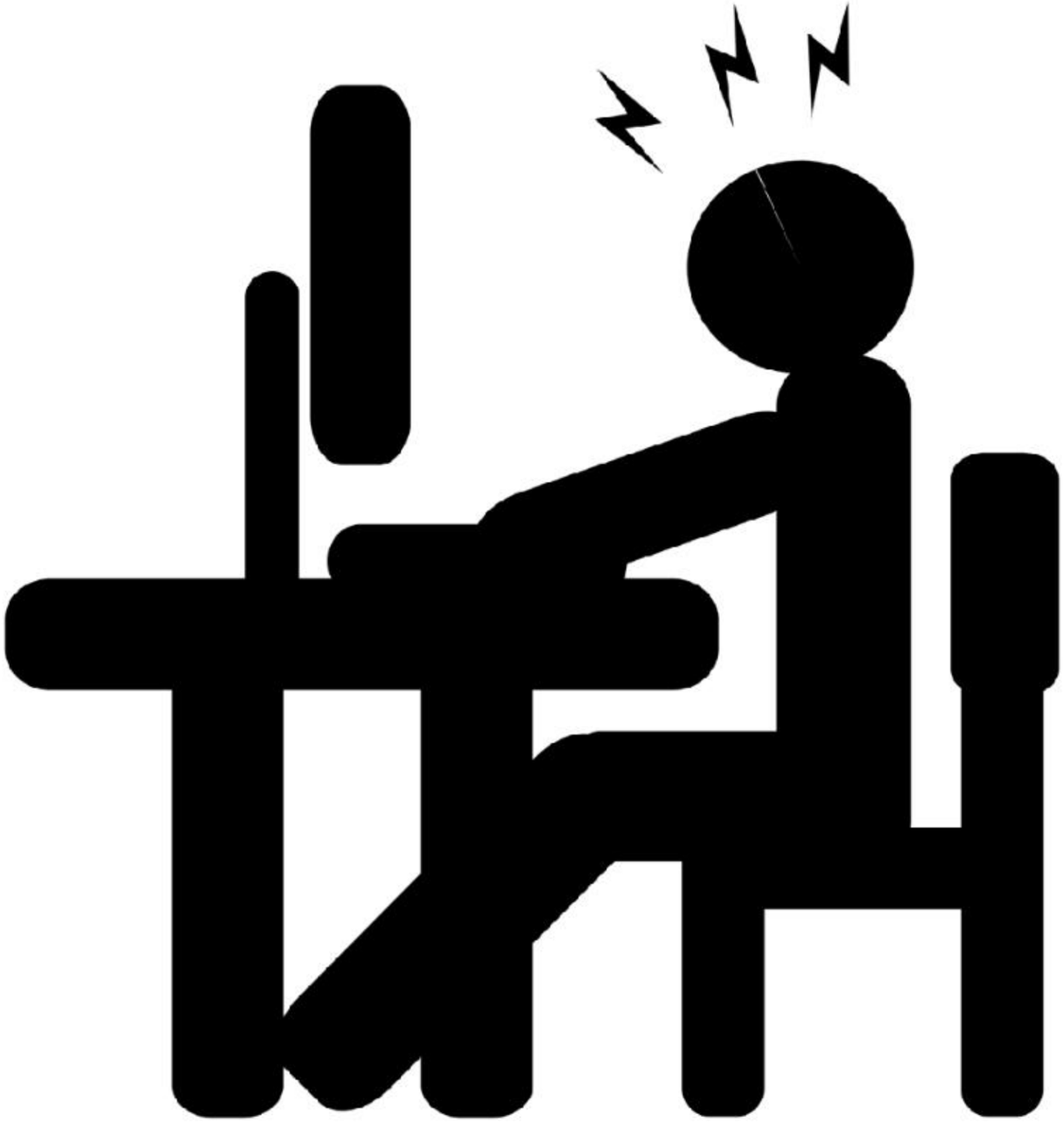


E-force (also autonomous)

https://eforce.cvut.cz

https://youtu.be/LCpDaRph_uA





https://gitlab.fel.cvut.cz/RPH-student-materials

http://cw.fel.cvut.cz/wiki/courses/b4b33rph/prednasky/start

https://www.root.cz/knihy/pro-git/

platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

```
> python3 variables_scope.py
```

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

variables_scope.py

platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

```
> python3 variables_scope.py
```

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

Co vytiskne řádek 13:

A: `2 14 3`

B: `1 14 3`

C: nastane chyba, `C` neznámé

D: nastane chyba už na řádku 12

variables_scope.py

platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

```
> python3 variables_scope.py
```

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

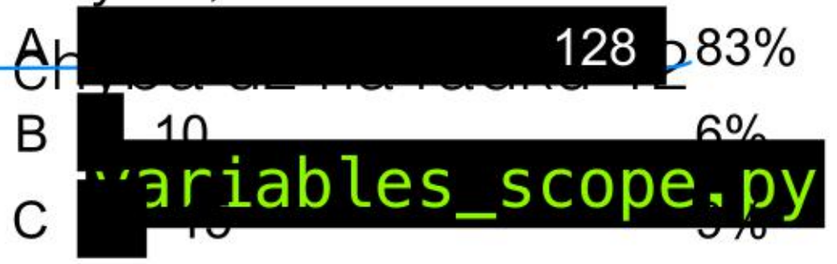
Co vytiskne řádek 13:

A: 2 14 3

B: 1 14 3

C: nastane chyba, C neznámé

D: nastane ch[redacted]



platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

> `python3 variables_scope.py`

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

Co vytiskne řádek 13:

A: `2 14 3`

B: `1 14 3`

C: nastane chyba, `C` neznámé

D: nastane chyba už na řádku 12

`variables_scope.py`

platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

```
> python3 variables_scope.py
```

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

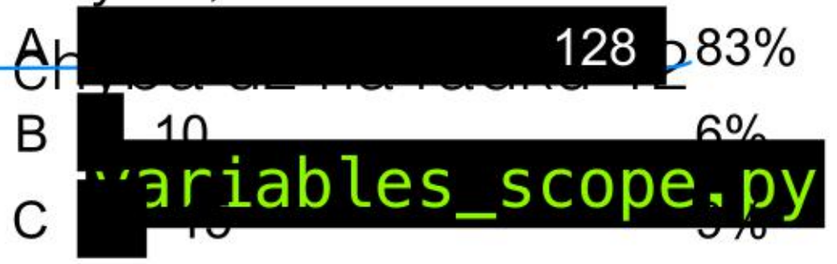
Co vytiskne řádek 13:

A: 2 14 3

B: 1 14 3

C: nastane chyba, C neznámé

D: nastane ch[redacted]



platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

```
> python3 variables_scope.py
```

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

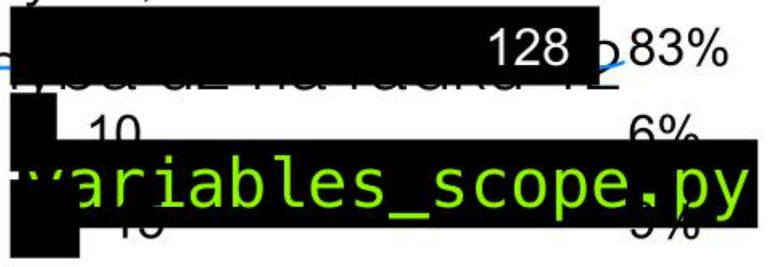
Co vytiskne řádek 13:

A: 2 14 3

B: 1 14 3

C: nastane chyba, C neznámé

D: nastane ch...



platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

> `python3 variables_scope.py`

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

Co vytiskne řádek 13:

A: `2 14 3`

B: `1 14 3`

C: nastane chyba, `C` neznámé

D: nastane chyba už na řádku 12

`variables_scope.py`

platnost proměnných

KONSTANTA = 1

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

> python3 variables_scope.py

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

Co vytiskne řádek 13:

A: 2 14 3

B: 1 14 3

C: nastane chyba, C neznámé

D: nastane chyba už na řádku 12

variables_scope.py

platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

> `python3 variables_scope.py`

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

Co vytiskne řádek 13:

A: `2 14 3`

B: `1 14 3`

C: nastane chyba, `C` neznámé

D: nastane chyba už na řádku 12

`variables_scope.py`

platnost proměnných

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

```
> python3 variables_scope.py
```

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

Co vytiskne řádek 13:

A: `2 14 3`

B: `1 14 3`

C: nastane chyba, `C` neznámé

D: nastane chyba už na řádku 12

variables_scope.py

platnost proměnných

```
> python3 variables_scope.py
```

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

Co vytiskne řádek 13:

A: `2 14 3`

B: `1 14 3`

C: nastane chyba, `C` neznámé

D: nastane chyba už na řádku 12

`variables_scope.py`

platnost proměnných

```
> python3 variables_scope.py
```

```
1  C = 3
2  a = 1
3
4  def my_function(x):
5      a = 9+C
6      return x+a
7
8  print(a)
9
10 if __name__ == "__main__":
11     a = 2
12     b = my_function(a)
13     print(a,b,C)
```

Co vytiskne řádek 8:

A: 12

B: 1

C: nic, nastane chyba za běhu

Co vytiskne řádek 13:

A: `2 14 3`

B: `1 14 3`

C: nastane chyba, `C` neznámé

D: nastane chyba už na řádku 12

variables_scope.py

function_vs_method.py

program structure - basic blocks

```
 1 import math
 2
 3 class MyClass:
 4     '''class for '''
 5     def __init__(self):
 6         '''MyClass constructor'''
 7         pass # nothing at the moment
 8
 9 def my_function(a,b):
10     '''compute sum a+b'''
11     pass # nothing at the moment
12
13 if __name__ == "__main__":
14     # actual program starts here
15     c = MyClass() # don't forget the parantheses! I will show!
16
```

function_vs_method.py

program structure - basic blocks

```
 1 import math                                                    imports
 2
 3 class MyClass:
 4     '''class for '''
 5     def __init__(self):
 6         '''MyClass constructor'''
 7         pass # nothing at the moment
 8
 9 def my_function(a,b):
10     '''compute sum a+b'''
11     pass # nothing at the moment
12
13 if __name__ == "__main__":
14     # actual program starts here
15     c = MyClass() # don't forget the parantheses! I will show!
16
```

function_vs_method.py

program structure - basic blocks

```
 1 import math
 2
 3 class MyClass:
 4     '''class for '''
 5     def __init__(self):
 6         '''MyClass constructor'''
 7         pass # nothing at the moment
 8
 9 def my_function(a,b):
10     '''compute sum a+b'''
11     pass # nothing at the moment
12
13 if __name__ == "__main__":
14     # actual program starts here
15     c = MyClass() # don't forget the parantheses! I will show!
16
```

imports

Definitions:
classes
functions

function_vs_method.py

program structure - basic blocks

```
 1 import math
 2
 3 class MyClass:
 4     '''class for '''
 5     def __init__(self):
 6         '''MyClass constructor'''
 7         pass # nothing at the moment
 8
 9 def my_function(a,b):
10     '''compute sum a+b'''
11     pass # nothing at the moment
12
13 if __name__ == "__main__":
14     # actual program starts here
15     c = MyClass() # don't forget the parantheses! I will show!
16
```

imports

Definitions:
classes
functions

main program

function_vs_method.py

program structure - basic blocks

```
 1 import math
 2
 3 class MyClass:
 4     '''class for '''
 5     def __init__(self):
 6         '''MyClass constructor'''
 7         pass # nothing at the moment
 8
 9 def my_function(a,b):
10     '''compute sum a+b'''
11     pass # nothing at the moment
12
13 if __name__ == "__main__":
14     # actual program starts here
15     c = MyClass() # don't forget the parantheses! I will show!
16
```

imports

Definitions:
classes
functions

main program

V krátkých ukázkách budeme někdy ukazovat jen kód bez hlaviček a spol.

funkce vs. metoda

```
 1 import math
 2
 3 class MyClass:
 4     '''class for '''
 5     def __init__(self):
 6         '''MyClass constructor'''
 7         pass # nothing at the moment
 8     def my_class_method(self):
 9         print('nothing to report')
10
11
12 def my_function(a,b):
13     '''compute sum a+b'''
14     pass # nothing at the moment
15
16 if __name__ == "__main__":
17     # actual program starts here
18     c = MyClass() # don't forget the parantheses! I will show!
19     c.my_class_method()
20
```

funkce vs. metoda

```
1  import math
2
3  class MyClass:
4      '''class for '''
5      def __init__(self):
6          '''MyClass constructor'''
7          pass # nothing at the moment
8      def my_class_method(self):
9          print('nothing to report')
10
11
12 def my_function(a,b):
13     '''compute sum a+b'''
14     pass # nothing at the moment
15
16 if __name__ == "__main__":
17     # actual program starts here
18     c = MyClass() # don't forget the parantheses! I will show!
19     c.my_class_method()
20
```

funkce vs. metoda

```
1  import math
2
3  class MyClass:
4      '''class for '''
5      def __init__(self):
6          '''MyClass constructor'''
7          pass # nothing at the moment
8      def my_class_method(self):
9          print('nothing to report')
10
11
12 def my_function(a,b):
13     '''compute sum a+b'''
14     pass # nothing at the moment
15
16 if __name__ == "__main__":
17     # actual program starts here
18     c = MyClass() # don't forget the parantheses! I will show!
19     c.my_class_method()
20
```

```
1 a = 0.1
2 b = 0.3
3 c = 3*a
4 d = b==c
```

```
1 a = 0.1
2 b = 0.3
3 c = 3*a
4 d = b==c
```

Po vykonání řádku 4 bude proměnná **d** rovna:

A: True

B: False

C: řádek 4 se nevykoná, skončí chybou

```
1 a = 0.1
2 b = 0.3
3 c = 3*a
4 d = b==c
```

0.3

Po vykonání řádku 4 bude proměnná **d** rovna:

A: True

B: False

C: řádek 4 se nevykoná, skončí chybou

není číslo jako číslo

```
1 a = 0.1
2 b = 0.3
3 c = 3*a
4 d = b==c
```

Po vykonání řádku 4 bude proměnná **d** rovna:

A: True

B: False

C: řádek 4 se nevykoná, skončí chybou

- vizualizace
- https://docs.python.org/3/tutorial/floatingpoint.html
- http://floating-point-gui.de/formats/binary/
- opatrnost při testování rovnosti (float) čísel
- pokud opravdu potřeba: `abs(a-b) < threshold`

`floating_point_surprise.py`

není číslo jako číslo

```
1 a = 0.1
2 b = 0.3
3 c = 3*a
4 d = b==c
```

Po vykonání řádku 4 bude proměnná **d** rovna:

A: True

B: False

C: řádek 4 se nevykoná, skončí chybou

$0.3 = 0.\ 10^{0} + 3.\ 10^{-1}$

$1/3 = 0{,}333\ldots$

- vizualizace
- https://docs.python.org/3/tutorial/floatingpoint.html
- http://floating-point-gui.de/formats/binary/
- opatrnost při testování rovnosti (float) čísel
- pokud opravdu potřeba: `abs(a-b) < threshold`

floating_point_surprise.py

decimal vs binary

Decimal (base 10)		Binary (base 2)
$1 \cdot 10^1 + 3 \cdot 10^0 = 13_{10}$	=	$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
$1 \cdot 10 + 3 \cdot 1 = 13_{10}$	=	$1101_2 = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1$

decimal-vs-binary-integers.png

decimal vs binary

Decimal (base 10)		Binary (base 2)
$1 \cdot 10^1 + 3 \cdot 10^0 = 13_{10}$	=	$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
$1 \cdot 10 + 3 \cdot 1 = 13_{10}$	=	$1101_2 = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1$

Decimal (base 10)		Binary (base 2)
$6 \cdot 10^{-1} + 2 \cdot 10^{-2} + 5 \cdot 10^{-3} = 0.625_{10}$	=	$0.101_2 = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$
$6 \cdot 1/10 + 2 \cdot 1/100 + 5 \cdot 1/1000 = 0.625_{10}$	=	$0.101_2 = 1 \cdot 1/2 + 0 \cdot 1/4 + 1 \cdot 1/8$

decimal-vs-binary-integers.png

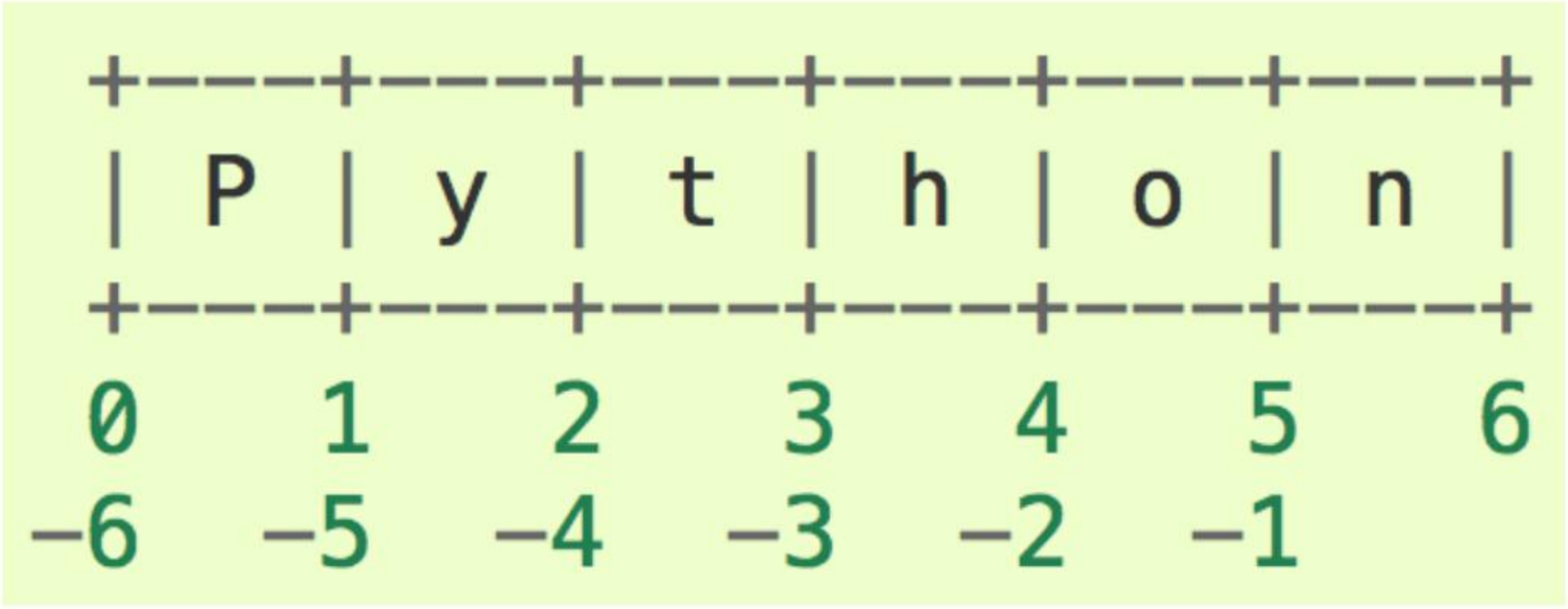
strings.py

řetězce neboli stringy

```
1 a = 'ahoj'
2 b = 'svete'
3 c = a+b
4 for i,item in enumerate(c):
5     print(i,'-',item)
6 banner = ['ahoj','svete']
7 for i,item in enumerate(banner):
8     print(i,'-',item)
9 for i,item in enumerate(banner):
10    for j,elem in enumerate(item):
11        print(i,'-',item,'***',j,':',elem)
```

visualizace

python indexing, slicing, …



dokončeme teď R-P-S
lepší hráč, s pamětí

- `player_skewed` některé tahy preferuje na úkor jiných
- nevíme které to jsou
- nejprve hrajme náhodně a uchovávejme soupeřovy tahy
- analyzujme historii tahů
- hrajme optimálně (využijme nedokonalost protihráče)

- `player_skewed` některé tahy preferuje na úkor jiných
- nevíme které to jsou
- nejprve hrajme náhodně a uchovávejme soupeřovy tahy
- analyzujme historii tahů
- hrajme optimálně (využijme nedokonalost protihráče)

is vs ==

```
1 a = [1,2,3]
2 b = [1,2,3]
3 c = a
4
5 d1 = a==b
6 d2 = a is b
7 d3 = a==c, a is c
```

is_vs_equal.py

- `player_skewed` některé tahy preferuje na úkor jiných
- nevíme které to jsou
- nejprve hrajme náhodně a uchovávejme soupeřovy tahy
- analyzujme historii tahů
- hrajme optimálně (využijme nedokonalost protihráče)