

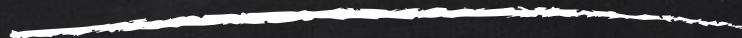
IB111 ÚVOD DO PROGRAMOVANÍ SKRZE PYTHON



Autor: Slavomír Krupa,
Text inšpirovaný Valdemarom Švábenským



INPUT





INPUT

★ Funkcia s jedným parametrom, ktorá načíta vstup

```
age = input("Enter your age: ")
```

- ★ Vypíše hodnotu svojho parametru a vráti hodnotu načítanú z klávesnice (zo shellu)
- ★ vráti premennú typu `string`
- ★ Chceme `int`? Ako skontrolovať či je v poriadku?



STRING

- ★ Pred pretypovaním vhodné overiť či je vôbec možné
- ★ Funkcie overujú aké znaky daný string obsahuje
- ★ `isalpha()` - len znaky abecedy
- ★ `isdecimal()` - len číslice (pretypovanie na `int`)
- ★ `isalnum()` - obsahuje len znaky abecedy a číslice

- ★ pretypovanie na `float`??



CONTENT CHECK

```
char = "aA"
```

```
a = char.isalpha()  
b = char.isdecimal()  
c = char.isalnum()
```

```
a = True  
b = False  
c = True
```



CONTENT CHECK

```
num = "7355608"
```



```
a = num.isalpha()  
b = num.isdecimal()  
c = num.isalnum()
```

```
a = False  
b = True  
c = True
```



CONTENT CHECK

```
mix = "A245b"
```

```
a = mix.isalpha()  
b = mix.isdecimal()  
c = mix.isalnum()
```

```
a = False  
b = False  
c = True
```



CONTENT CHECK

```
flt = "25.4"
```

```
a = flt.isalpha()  
b = flt.isdecimal()  
c = flt.isalnum()
```

```
a = False  
b = False  
c = False  
d = True
```

```
d = flt.replace(".", "", 1).isdecimal()
```




SEARCH



LINEAR SEARCH

- ★ Ak n je počet prvkov zoznamu, algoritmus spraví najviac n porovnaní (teda skontroluje najviac n prvkov)
 - S rastúcim n predpokladáme lineárny nárast počtu krokov algoritmu
 - Pre 1000000 prvkov maximálne 1000000 krokov

Key	List
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8



BINARY SEARCH

start

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

guess: 5

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

$x < 5$

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

guess: 2

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

$x > 2$

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

guess: 3

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

$x > 3$

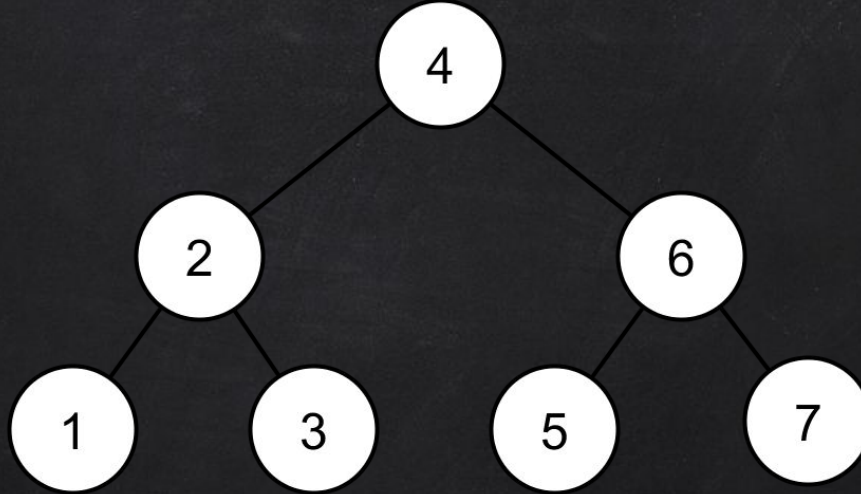
1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

finish

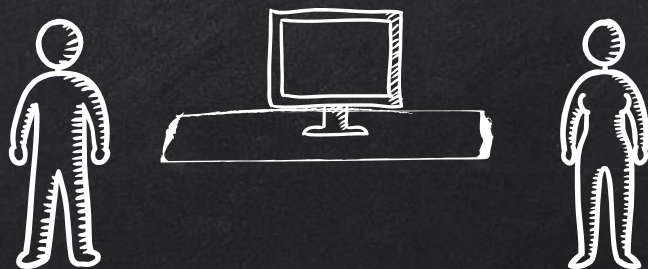
1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



BINARY SEARCH



1	2	3	4	5	6	7
---	---	---	---	---	---	---



?PAIR PROGRAMMING?



HRA

- ★ Máme dvoch hráčov, A a B
- ★ A si myslí prirodzené číslo od 1 do n
- ★ B tipuje čísla
- ★ A odpovedá: moje číslo je menšie/rovné/väčšie ako tvoj tip
- ★ Akú stratégiu má B použiť, aby čo najrýchlejšie uhádol?



ÚLOHA 1

- ★ Hráč A = počítač („myslí“ si číslo od 1 do n)
- ★ Hráč B = používateľ (tipuje čísla)
- ★ Nemusíte riešiť ošetrenie vstupu
- ★ (príklad na ďalšom slide...)



ÚLOHA 1

```
>>> guess_number(10)
Enter your guess( or 0 to end) :
5
My number is lower
Enter your guess( or 0 to end) :
3
My number is lower
Enter your guess( or 0 to end) :
1
My number is greater
Enter your guess( or 0 to end) :
2
You won on 4 attempt.
```




ÚLOHA 2

- ★ Hráč A = používateľ (myslí si číslo od 1 do n)
- ★ Hráč B = počítač (pýta sa, či je hľadané číslo menšie, rovné alebo väčšie ako jeho tip)
- ★ Nemusíte riešiť ošetrovanie vstupu
- ★ (príklad na ďalšom slide...)



ÚLOHA 2

```
>>> guess_number_pc(10)
Think of number from 1 to 10
Is your number lower (-1), equal (0) or greater (1) than 5 ?
1
Is your number lower (-1), equal (0) or greater (1) than 8 ?
-1
Is your number lower (-1), equal (0) or greater (1) than 6 ?
1
Your number is 7
```



ÚLOHA 3

- ★ Naprogramujte funkciu pre binárne vyhľadávanie (bez slajdov z prednášky)
- ★ Fungujúcu aj pre nezoradené zoznamy
 - Použite sort

```
>>>binary_search(7, [1, 4, 5, 7, 9])
True
>>>binary_search(8, [1, 4, 5, 7, 9])
False
>>>binary_search(7, [7, 5, 9])
True
```



ÚLOHA 4

- ★ Do funkcie z Úlohy 3 pridajte počítadlo krokov (iterácií hlavného cyklu)
- ★ Rátajte počet iterácií pre veľké vstupné zoznamy
- ★ Vytvorte si samostatnú funkciu, ktorá vygeneruje zoznam k čísel a vráti
- ★ Analyzujte výsledky ako v DÚ 2



ÚLOHA 5

- ★ Naprogramujte funkciu pre lineárne vyhľadávanie
- ★ Pridajte počítadlo iterácií
- ★ Analyzujte výsledky ako v Úlohe 4



ÚLOHA 1*



Rozšířte úlohu 1 o kontrolu vstupu

```
>>>guess_number(10)
Enter your guess( or 0 to end) :
adad
Wrong input enter again
Enter your guess( or 0 to end) :
565a
Wrong input enter again
Enter your guess( or 0 to end) :
22
My number is lower
Enter your guess( or 0 to end) :
```



ÚLOHA 2*

- ★ Hráč A = počítač („myslí“ si číslo od 1 do n a odpovedá, či je hľadané číslo menšie, rovné alebo väčšie ako tip B)
- ★ Hráč B = počítač (tipuje čísla)

CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Unsplash](#)