

Boolean operators.  
Statements. While loop.



# Какво научихме до сега?

1. Стойности и изрази.
2. Променливи.
3. Типове и оператори.
4. Функции / потребителски вход.
5. Да пускаме код в REPL.
6. Какво е булева променлива и булев израз?
7. Syntax на if структурата.



# Стойности и изрази.

Изразите се свеждат  
до стойности!

```
>>> 5
5
>>> 5 + 5
10
>>> "Hack" + "Bulgaria"
"HackBulgaria"
```



# В 1 израз могат да участват:

- Променливи
- Стойности
- Скоби
- Оператори
- Функции
- Валиден синтаксис!

```
>>> 1 + (randint(1, 6) + total_sum) * times
```



# Променливи

1. Дават име на израз или стойност.
2. Могат да променят своята стойност.
3. Могат да променят своята стойност чрез старата си стойност.

```
sum = 0
repeat = 5

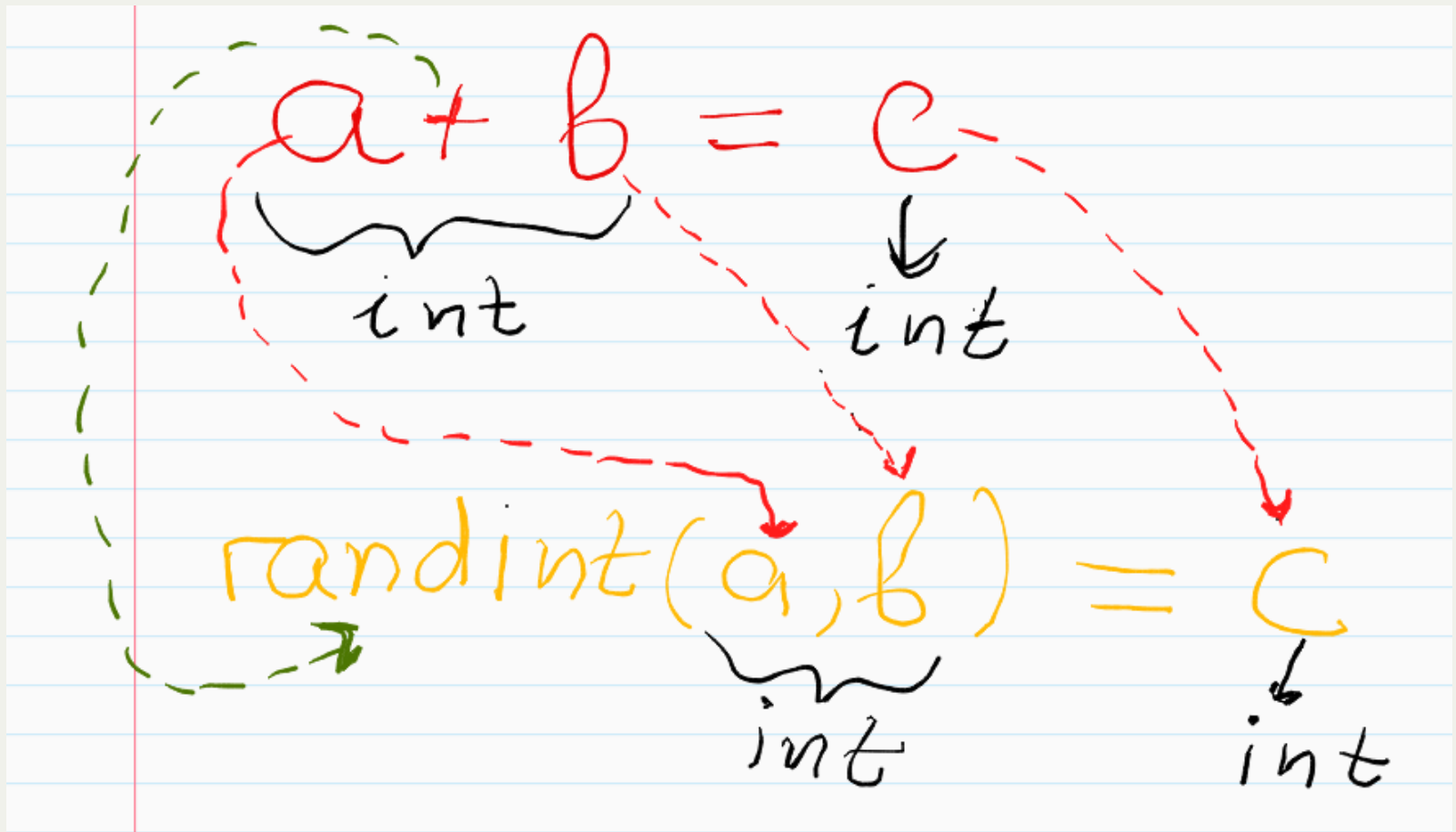
while repeat > 0:
    sum = sum + repeat
    repeat = repeat - 1
```



# Типове и оператори.

1. Типовете са "кошниците", които групират отделните стойности в нашия език.
2. Операторите работят благодарение на съществуването на типовете.
3. Операторите на 2 операнди се наричат бинарни оператори.





Операторите може да бъдат разгледани като функции!

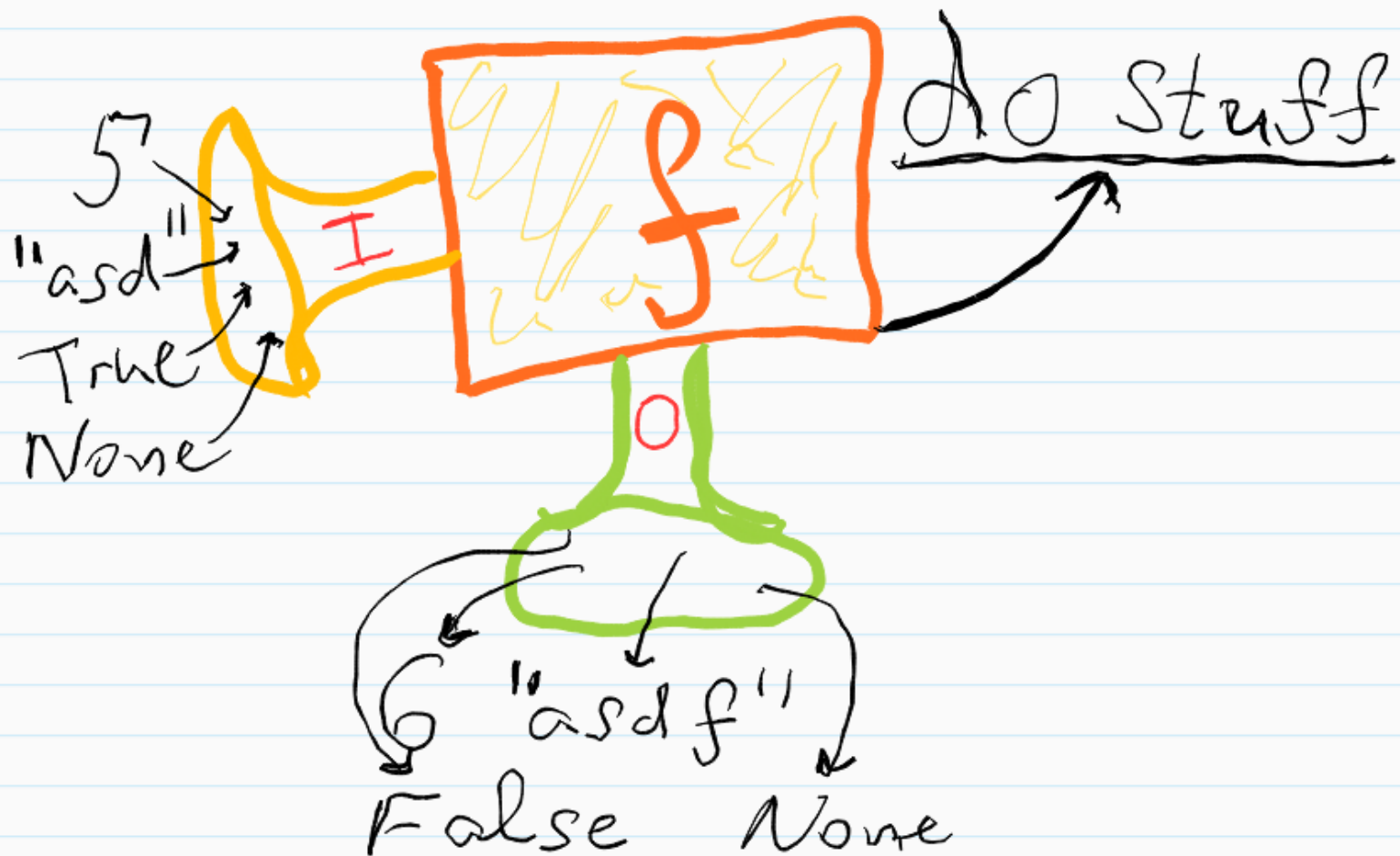


# Функции

1. Представяват парче код, скрито зад името на функцията.
2. Вършат точно определена работа.
3. Приемат вход (аргументи) и дават изход (връщат резултат)
4. Функциите се "викат", като им се дадат нужните аргументи.
5. Функциите са като израз – ако ги извикаме, получаваме резултат.







# Потребителски вход чрез input

1. Дава възможност да въвеждаме данни извън програмата.
2. **Потребителският вход винаги е string!**
3. "Спира" изпълнението на програмата, докато не бъде въведен вход.
4. Приема аргумент, string, който ни подсказва какво да се въведе.



# Текстови файлове и REPL

1. Пишем код в текстови файлове.
2. Даваме нашия код на специална програма (компилятор), която чете нашия код и го изпълнява.
3. Read Evaluate Print Loop ни позволява да изпълняваме парчета код, които не са в цяла програма.
4. F5 в IDLE изпълнява нашата програма в REPL-а



# Булев тип и Булев израз.

1. **Булевият тип** има 2 стойности – True и False
2. **Булев израз** е израз, който се оценява до True или False
3. Имаме оператори като == и !=, които връщат булев тип.
4. Използваме булеви изрази в конструкции като If



# Чрез if взимаме решения в нашата програма.

1. Питаме неща като: **"Ако тази променлива има такава стойност, направи нещо. Иначе направи нещо друго."**
2. Може да избираме сред повече от 2 избора.  
**Тогава ползваме elif.**



# Една if конструкция може да бъде:

Само if

```
age = input("Enter age?")  
age = int(age)  
  
if age < 18:  
    print("You are not allowed here")
```



# Една if конструкция може да бъде:

if с else

```
if age < 18:  
    print("You are not allowed here")  
else:  
    print("Hello there!")
```



# Една if конструкция може да бъде:

С множество If-a без else

```
if age < 18:  
    print("You are not allowed here")  
elif age == 18:  
    print("You barely made it here!")  
elif age == 33:  
    print("Anything special about that age?")
```





# Една if конструкция може да бъде:

С множество If-a с else

```
if age < 18:  
    print("You are not allowed here")  
elif age == 18:  
    print("You barely made it here!")  
elif age == 33:  
    print("Anything special about that age?")  
else:  
    print("You are > 18 but not 18 and 33")
```



# Statements.

Expressions се оценяват до  
някаква стойност.

1. Statements нямат стойност.
2. Те се използват за контролиране на програмата.
3. Те въвеждат странични ефекти.



# Statements.

1. Създаване на променлива е statement.
2. If конструкцията е statement.
3. While конструкцията е statement.
4. **Една програма се състои от: променливи + изрази + statements.**

```
# Variable assingment is a statement  
a = 5
```



# Примерна програма

```
contents = open("alumni.txt").read().split("\n")
csv = []

for person in contents:
    person = person.replace(" - ", ",")
    csv.append(person)

csv_content = "\n".join(csv)
file = open("alumni.csv", "w")
file.write(csv_content)
```



# Оператори за комбиниране на булеви стойности.

1. Логическо и се – **and**
2. Логическо или – **or**
3. Логическо отрицание – **not**



# a and b

1. Питаме дали две условия са изпълнени едновременно
2. Връща True само при **a = True** и **b = True**
3. Всички други случаи – False

```
if age > 18 and money >= 50:  
    print("You can enter the club")
```



# a or b

1. Питаме дали поне едно от условията е изпълнено.
2. Връща False само при **a = False** и **b = False**
3. Всички други случаи – True

```
if talking_on_the_phone or without_seatbelt:  
    print("Приятно ми е, полицай Георгиев!")  
    money = money - 20
```



# not a

1. Отрицание.
2. Превръща True във False
3. Превръща False във True

```
if not sleeping:  
    print("You are awake!")
```





# Оператор in

1. Проверява дали string се среща в друг string.
2. Проверява дали елемент се среща в списък.
3. Отрицанието му е **not in**
4. **Връща True или False**

```
if "Rado" in "RadoRado":  
    print("I found the Rado")
```



# Още 1 пример с in

```
name = input("What is your name?")
search = "Yordanov Georgiev"

if search not in name:
    print("All is fine. You can continue")
else:
    print("Please step outside the car!")
```



# Решете задачи от 3-And- Or-Not-In-Problems



# While loop – да повторим 1 действие няколко пъти.

Имаме следния проблем – направихме играта със заровете. Обаче има промяна в правилата – трябва да хвърлят 10 пъти, за да определим победителя!



# Възможни решения

1. Да си повторим кода 10 пъти, с нужните проверки.
2. Да изнесем играта във функция и да я извикаме 10 пъти.
3. Трябва да има по-хубав начин?



# Повтаряй, докато е изпълнено дадено условие.

1. Брояч на хвърлянията = 0
2. Докато брояча не стане 10.
3. Хвърляй зарче и помни кой е победил.
4. Накрая обяви победителя.



```
from random import randint

sides = input("Enter dice sides: ")
sides = int(sides)

player1_name = input("Enter player1 name: ")
player2_name = input("Enter player2 name: ")

player1_dice_roll = randint(1, sides)
player2_dice_roll = randint(1, sides)

print(player1_name + " rolled: " + str(player1_dice_roll))
print(player2_name + " rolled: " + str(player2_dice_roll))

if player1_dice_roll == player2_dice_roll:
    print("Tie!")
elif player1_dice_roll > player2_dice_roll:
    print("The winner is: " + player1_name)
else:
    print("The winner is: " + player2_name)
```

# while синтаксис – като if без else

```
# while.py  
  
counter = 0  
  
while counter <= 5:  
    print(counter)  
    counter = counter + 1
```





# while синтаксис – като if без else

```
while bool_expr:  
    print("Runs while bool_expr is True")  
    print("The first False breaks the cycle")
```



```
while False:  
    print("No one reaches me ;(")
```

```
while True:  
    print("You are forever under my control!")
```

```
counter = 0  
  
while True:  
    counter = counter + 1  
  
    if counter == 10:  
        break  
  
print(counter)
```



# Един цикъл може:

1. Никога да не се изпълни.
2. Да се върти безкрайно.
3. Да бъде прекъснат с ключовата дума **break**
4. **Да ни спаси живота!**



Решете задачите от 4-  
While-Loop!

