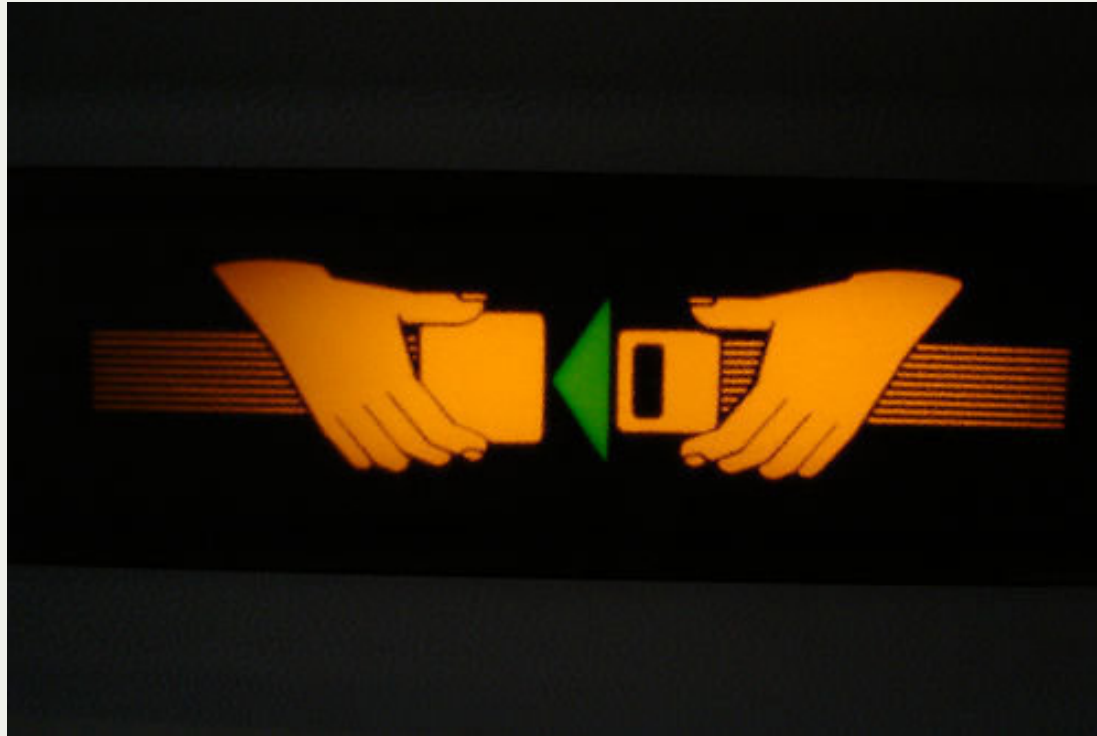


# Types. Input/Output. Booleans & If statement.





Първо ще загреем с  
проверка от предния път.

Извадете си по 1 лист ;)



# Решение на домашното.

Има много идеи, които с напредването на курса  
ще разглеждаме и ще се връщаме към това  
домашно.



5те основни values, които  
видяхме в Python, се  
характеризират по още 1  
начин.



5

5.0

"Python"

True

False

None



# Всеки основен value има свой тип.

Тоест, всяка стойност носи определен тип.

Типовете, по-известни като `types`, играят ключова роля в света на програмирането.



**5 - Integer**

**5.0 - Float**

**"Python" - String**

**True - Boolean**

**False - Boolean**

**None - NoneType**





# Няколко много важни неща!

1. Всяка стойност има тип.
2. Всеки израз се свежда до стойност  
=>

**Всеки израз има тип.**

3. Променливите?
4. Променливите са или стойност или израз  
=>

**Променливите имат тип.**



# Типовете служат за обобщение на отделните стойности.

5 e int

6 e int

1231231 e int



# Всичко е свързано!



Операторите работят  
благодарение на  
типовете.

$$a + b$$



$a + b$  дава резултат, когато:

1.  $a$  и  $b$  са `int` или `float` (числа)
2.  $a$  и  $b$  са `strings` (текст)
3.  $a$  и  $b$  са списъци (по-нататък)
4. Всички други случаи – `TypeError`



$a * b$  дава резултат, когато:

1.  $a$  и  $b$  са `int` или `float`.
2.  $a$  е `string`,  $b$  е `int`
3.  $b$  е `string`,  $a$  е `int`
4.  $a$  е списък (по-нататък),  $b$  е `int`
5.  $a$  е `int`,  $b$  е списък (по-нататък)
6. Всичко друго – `TypeError`



# Статично типизиран език

vs.

# Динамично типизиран език



# Статично типизирани езици

```
// example in Java  
String name = "Maria";  
int age = 23;
```





# Динамично типизирани езици

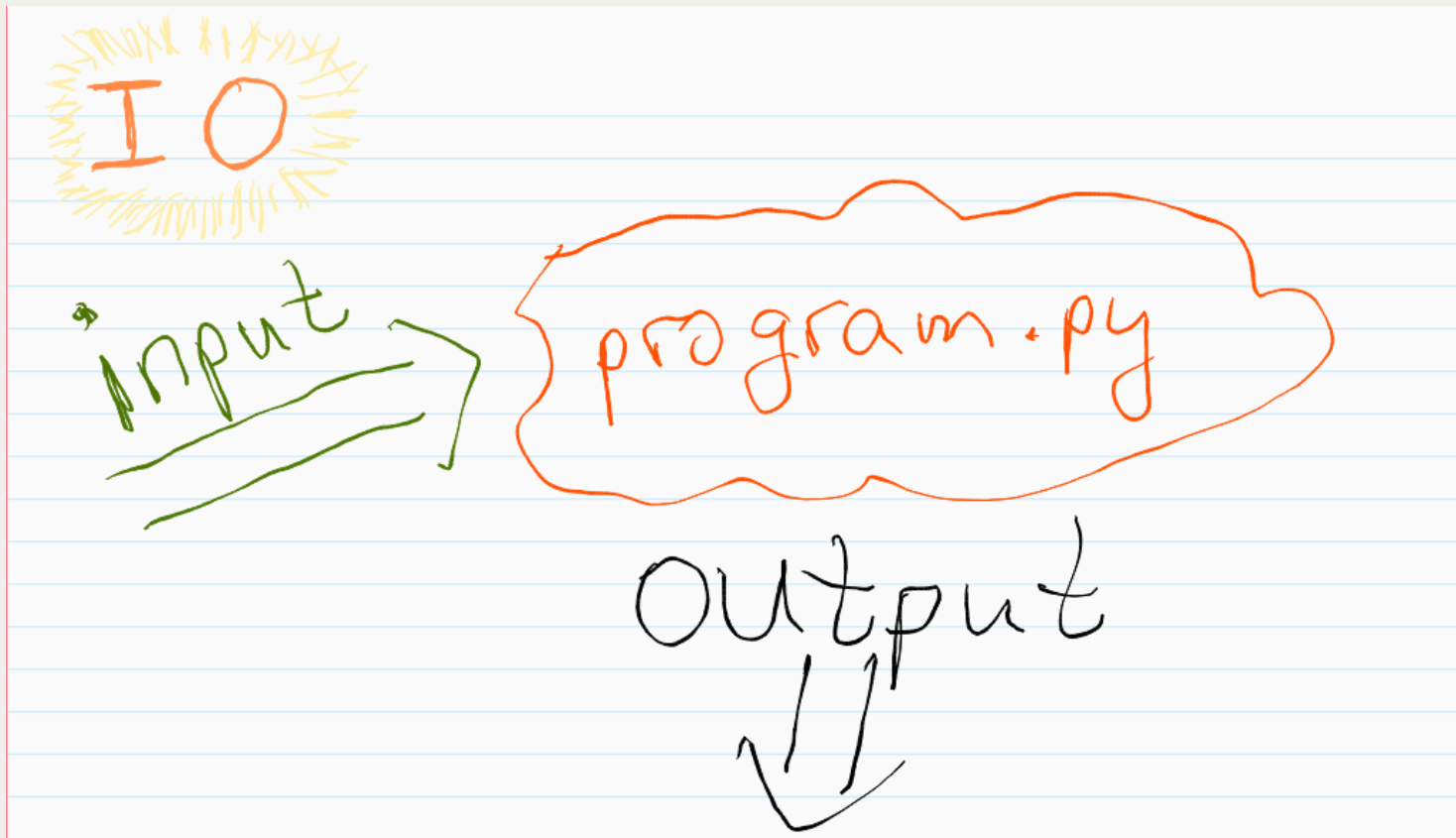
```
# example in Python  
name = "Maria"  
age = 23
```



Глътнете малко въздух,  
правим почивка!



# Вход и Исход.



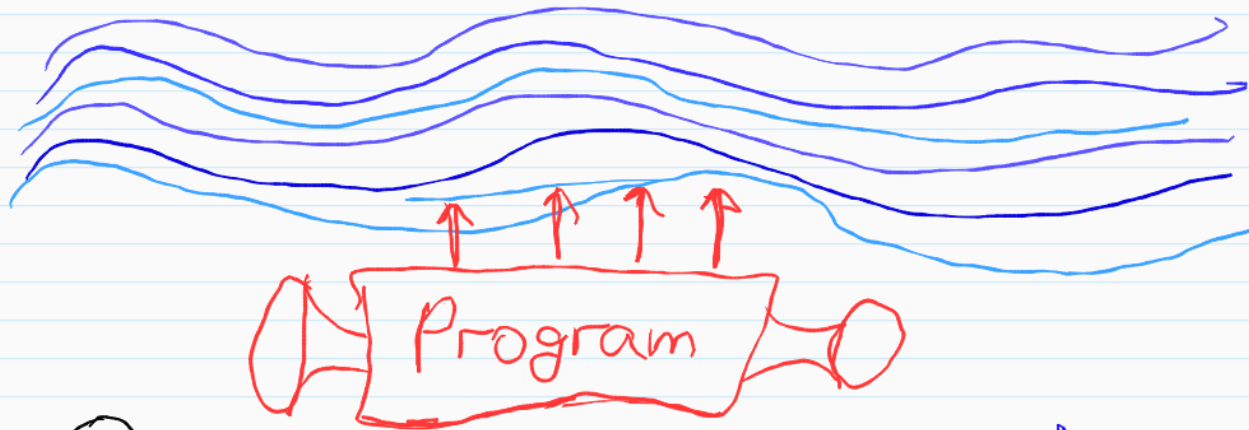
# Нашите програми имат потребители.

Потребителите ни дават вход, а ние им даваме  
ИЗХОД.



# Става на въпрос за

Information flow



- ① Read ▶▶▶
  - ② Process ▶▶▶
  - ③ Output ▶▶▶
- Data!



# Нещата са много Zen:

1. Четем данни.
2. Обработваме данни.
3. Даваме данни като резултат.

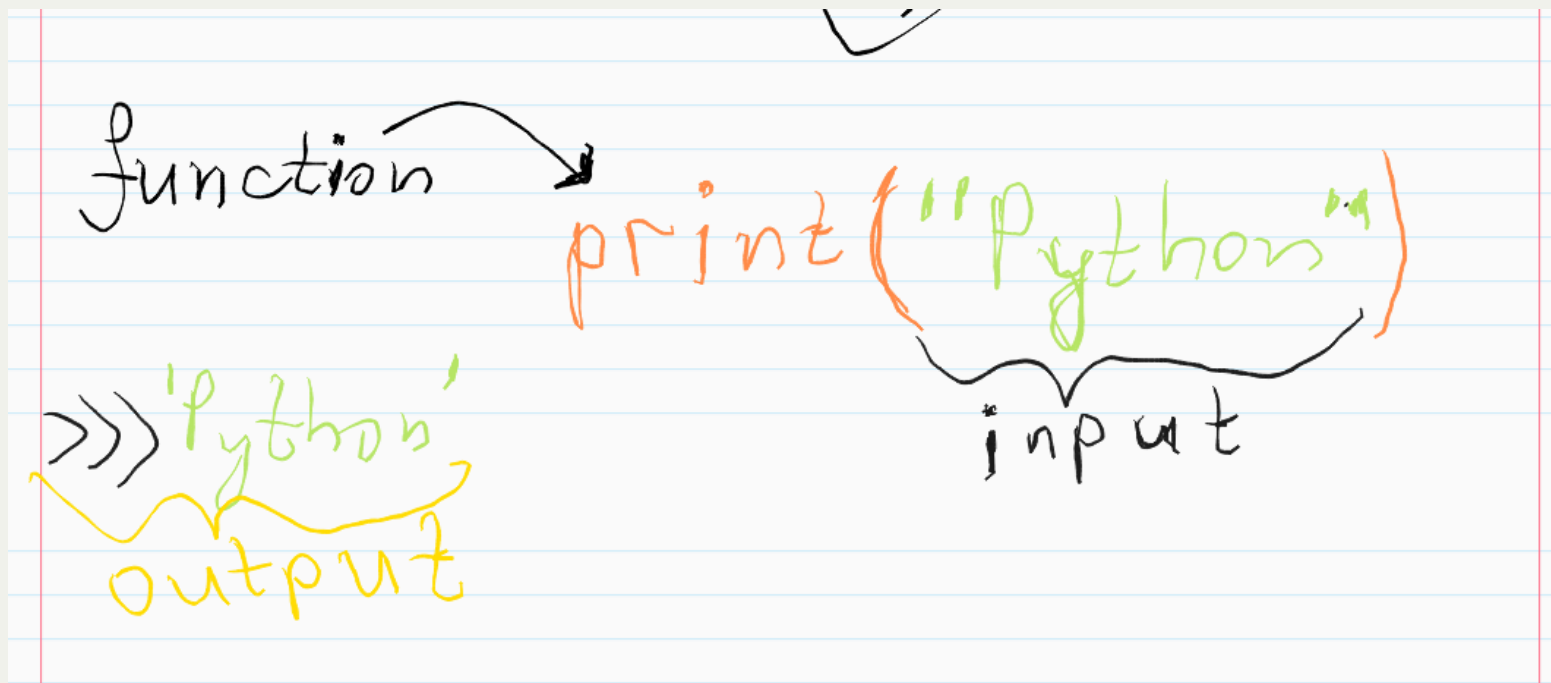


# Примерна програма за ВХОД/ИЗХОД:

```
# hithere.py  
  
name = input("What is your name?")  
print("Hi there!")  
print(name)
```



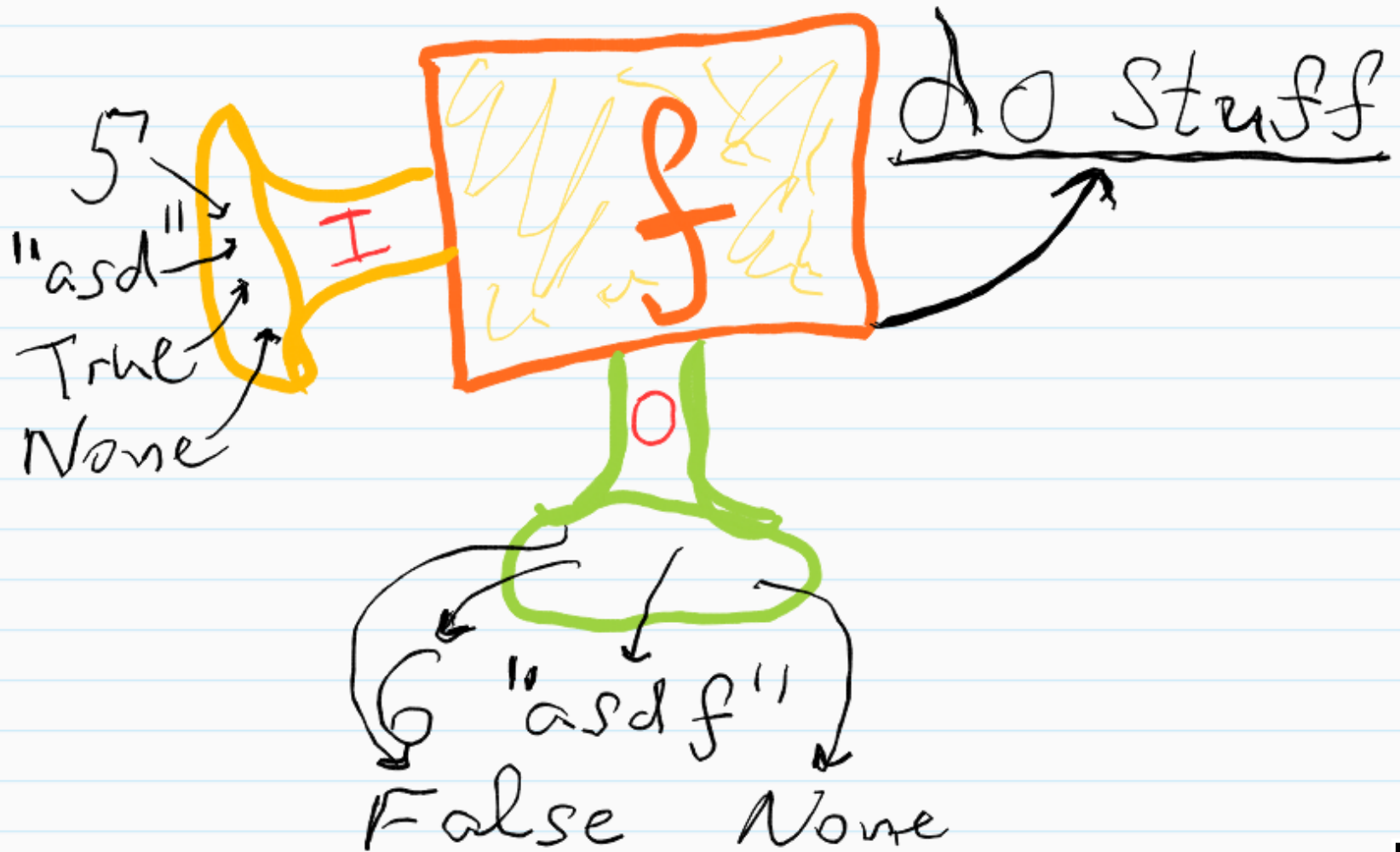
# print() и input() са функции.





Функциите са кутия,  
която върши определена  
дейност и връща  
резултат.





# Функциите са като изразите. Връщат определен тип (стойност)

```
# dice_roll.py

from random import randint

dice = randint(1, 6)

print("I have thrown the dice of destiny. Here is the result:")
print(dice)
```



Когато четем вход от потребител чрез `input`, то **той винаги е `string`!**

```
# what.py

a = input("Enter number for a:")
b = input("Enter number for b:")

sum = a + b

print("The sum between a and b is:")
print(sum) # Concatenated a and b
```



Функцията `int()` взима `string` и връща `int`, ако е  
ВЪЗМОЖНО.

```
a = "10"  
print(a + a) # "1010"  
  
b = int(a)  
print(b + b) # 20
```



# Задачи!

Решете задачите от 1-  
10-Simple-Problems.



Булевият тип.

Запознайте се с **True** и  
**False**.



Булевият тип в  
програмирането ни  
позволява да "взимаме  
решения" с нашия код.

"Ако се случи това, направи  
онова. Иначе друго"





Казвахме, че изразите  
(expressions) се свеждат  
до стойност, която има  
даден тип.



В int, как комбинираме  
отделните стойности в  
изрази?

Чрез оператори!



Операторите == и !=  
комбинират различни  
типове до булеви  
променливи.

```
5 == 5 # True
5 != 5 # False
"Python" == "Python" # True
"Python" != "C++" # False
```



# **a == b** работи по следния начин:

1. Ако **a** и **b** са различни типове, **False**
2. Ако **a** и **b** не съвпадат като стойност, **False**
3. **True** в краен вариант.



# **a != b** работи по следния начин:

1. Ако a и b са различни типове, True
2. Ако a и b не съвпадат като стойност, True
3. False в краен вариант.



**Булев израз наричаме  
такъв израз, който се  
свежда до **True** или  
**False**.**



С if конструкцията,  
Булевите променливи и  
операторите за  
комбиниране, взимаме  
решения в нашите  
програми!









# if има следния синтаксис:

```
if boolean_expression:  
    # Do something when boolean_expression is True  
else:  
    # Do something when boolean_expression is False
```

```
a = 5  
b = 5  
  
if a + b == 10:  
    print("5+5 is 10")  
else:  
    print("5+5 is not 10")
```



# if има следния синтаксис:

1. `if boolean_expression:`
2. **На 4 шпации разстояние!!**
3. `else:`
4. **На 4 шпации разстояние!!**
5. **Не забравяйте :**



Много често проверяваме  
дали дадена стойност е  
това, което очакваме:

```
answer = input("Enter me a number")
answer = int(answer)

if answer == 42:
    print("You know the secret of the universe!")
else:
    print("Nothing interesting about:")
    print(answer)
```



Ами ако искаме да  
проверим дали това,  
което очакваме може да е  
повече от 1 стойност?



```
# choice.py

from random import randint

print("Enter 1 for a joke")
print("Enter 2 for a random number")
print("Enter 3 for for nothing")

number = input("What do you choose?")

if number == 1:
    print("Something funny here")
elif number == 2:
    random_number = randint(1,100)
    print("A random number between 1 and 100:")
    print(random_number)
elif number == 3:
    print("Nothing!")
```

Всичко наред ли е?



```
# choice.py

from random import randint

print("Enter 1 for a joke")
print("Enter 2 for a random number")
print("Enter 3 for for nothing")

number = input("What do you choose?")
number = int(number)

if number == 1:
    print("Something funny here")
elif number == 2:
    random_number = randint(1,100)
    print("A random number between 1 and 100:")
    print(random_number)
elif number == 3:
    print("Nothing!")
```



# Оператори, които се използват за съставяне на булев израз:

1. ==

2. !=

3. >

4. <

5. >=

6. <=



# Ако човек е по-малък от 18 години:

```
# age.py

age = input("What is your age?")
age = int(age)

if age < 18:
    print("This is a retracted area.")
```





Това е края.

Решете задачите от 2-If-  
Elif-Else-Simple-  
Problems

