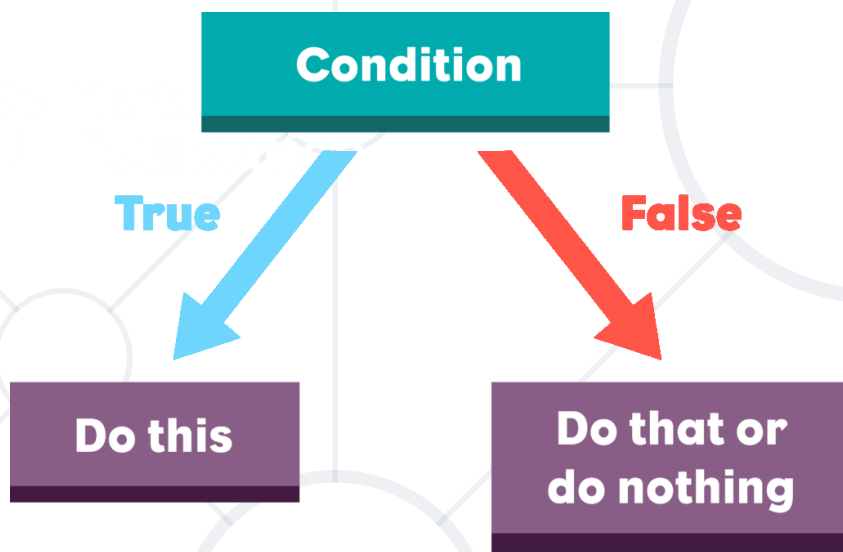


# Условни конструкции

Логически изрази и проверки. Условна конструкция if-else



СофтУни

Преподавателски екип



Software  
University



SoftUni  
Foundation



Software University

<http://softuni.bg>

# Имате въпроси?

sli.do

#pb-jan

1. Проверки
2. Логически изрази и проверки
  - Оператори за сравнение.
3. Условни конструкции
4. Серия от проверки
5. Живот на променлива
6. Дебъгване
7. Решаване на изпитна задача





**Преговор**

1. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
print('a' + 'b')
```

ab

Error

ba

195

1. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
print('a' + 'b')
```

ab

Error

ba

195

2. Какъв е типът на променливата:

```
number = "1000"
```

str

int

float

2. Какъв е типът на променливата:

```
number = "1000"
```

str

int

float



3. Как се нарича долепването на два текста (низа)?



Събиране

Конкатенация

Кулминация

Съединяване

3. Как се нарича долепването на два текста (низа)?



Събиране

Конкатенация

Кулминация

Съединяване

4. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
print(10 % 3)
```

10

1

0

3

4. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
print(10 % 3)
```

10

1

0

3

5. Каква стойност държи променливата **result**:

```
a = 5  
b = 2  
result = a / b
```

2

7

2.5

1

5. Каква стойност държи променливата **result**:

```
a = 5  
b = 2  
result = a / b
```

2

7

2.5

1

6. Какъв би бил резултатът, ако се опитаме да изпълним следната команда:

```
print(1 + 1 + "4" + 2 + 1)
```

Error

243

9

2421

6. Какъв би бил резултатът, ако се опитаме да изпълним следната команда:

```
print(1 + 1 + "4" + 2 + 1)
```

Error

243

9

2421





**Логически изрази и проверки**

**Оператори за сравнение**

# Оператори за сравнение



Оператор	Означение	Работи за
Равенство	==	числа, дати, други сравними типове
Различно	!=	
По-голямо	>	
По-голямо или равно	>=	
По-малко	<	
По-малко или равно	<=	

- В програмирането можем да сравняваме стойности
  - Резултатът от логическите изрази е **True** или **False**

```
a = 5
b = 10
print(a < b)           # True
print(a > 0)           # True
print(a > 100)         # False
print(a < a)           # False
print(a <= 5)          # True
print(b == 2 * a)      # True
```

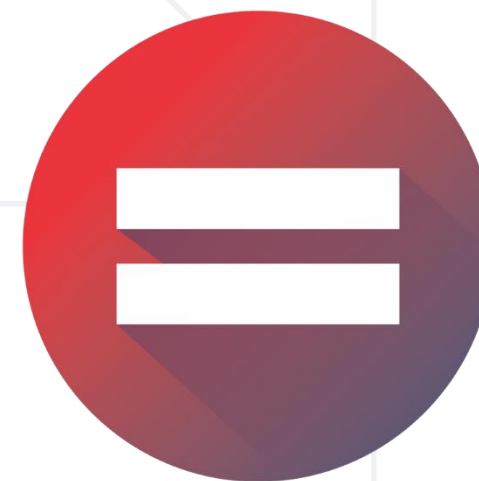


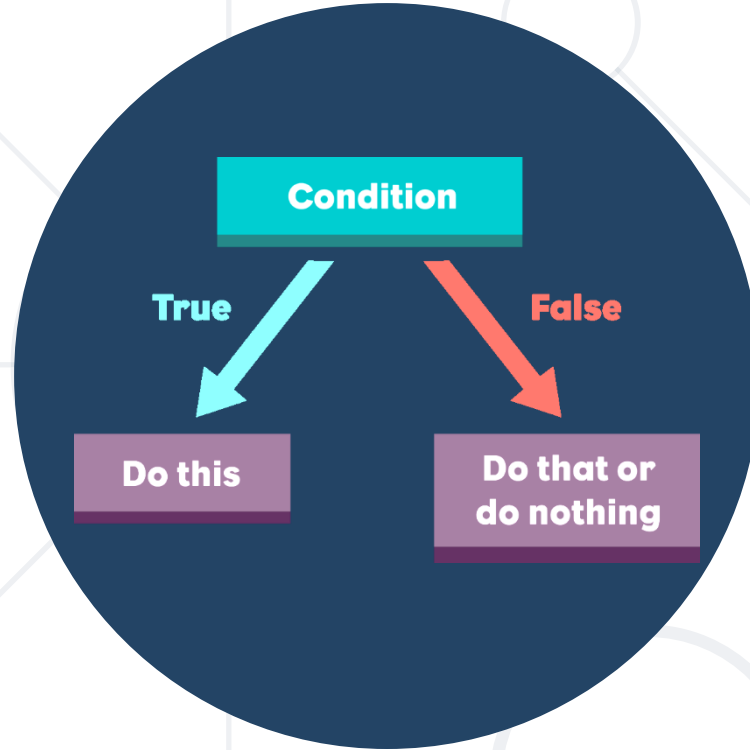
- Сравняване на текст чрез оператор за равенство (==)

```
a = 'Example'  
b = a  
print(a == b)      # True
```

```
a = input()  
b = input()  
print(a == b)      # True
```

Въвеждане на  
еднаква стойност






# Условни конструкции

## Прости проверки

# Прости проверки

- Често проверяваме условия и извършваме действия според резултата



Условие  
(булев израз)

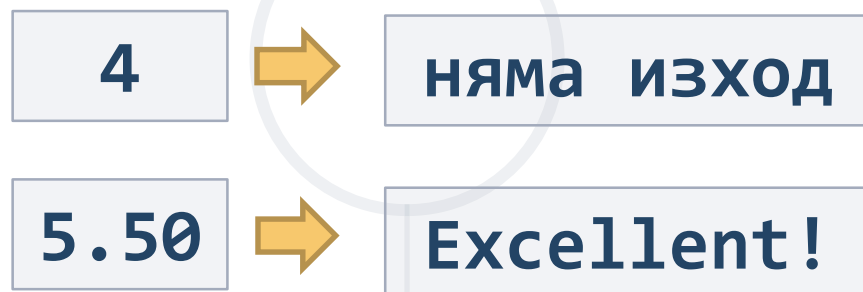
```
if ...:  
    # код за изпълнение
```

Код за изпълнение при  
вярност на условието

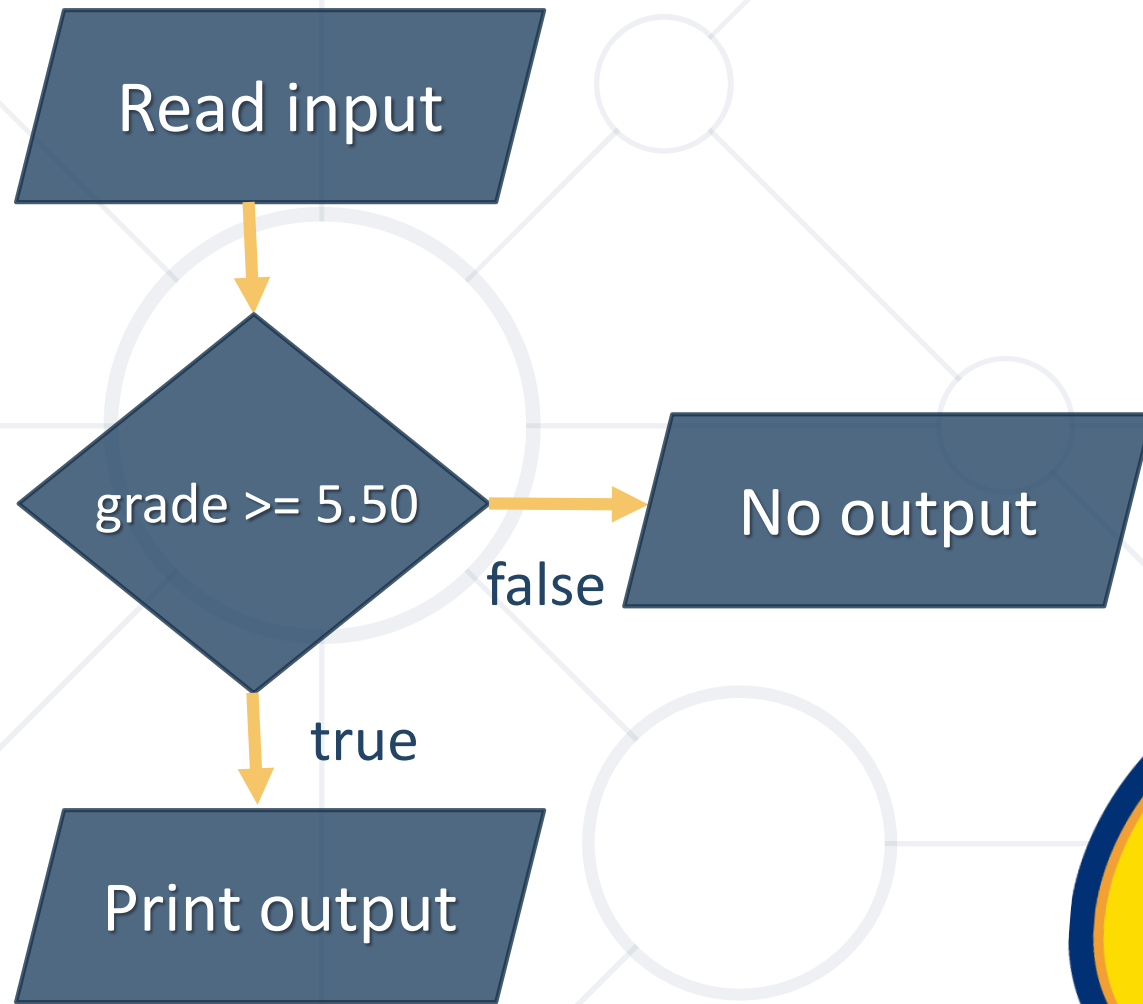
- Резултатът е **True** или **False**

- Напишете **програма**, която:
  - **Чете** оценка (**число**), въведена от потребителя
  - **Проверява** дали е отлична
  - **Извежда** "Excellent!", ако оценката е по-голяма или равна на 5.50

- Пример:



236



6





# Прости проверки – if-else

- При **невярност** (False) на условието, можем да изпълним други действия – чрез **else** конструкция



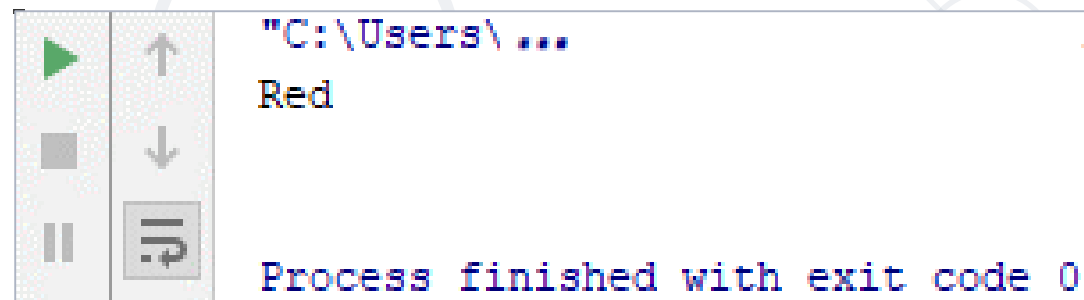
```
if ...:  
    # код за изпълнение  
else:  
    # код за изпълнение
```

Код за изпълнение  
при невярност на  
условието

- Табулациите въвеждат блок от код (група команди)
  - Изпълнява се редът, който отговаря на условието

```
color = 'red'  
if color == 'red':  
    print('Red')  
else:  
    print('Yellow')  
    print('bye')
```

Извежда се "Red"



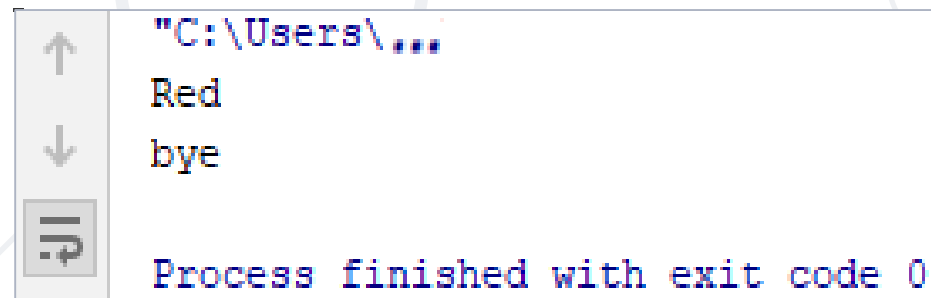
```
"C:\Users\ ...  
Red  
  
Process finished with exit code 0
```

- Без табулации ще се изпълнява и **последният** ред

```
color = 'red'  
if color == 'red':  
    print('Red')  
else:  
    print('Yellow')  
print('bye')
```

Изпълняват се  
редовете отговарящи  
на условието

Изпълнява се винаги – не е  
част от if/else конструкцията



```
"C:\Users\...  
Red  
bye  
  
Process finished with exit code 0
```

# По-голямото число – условие

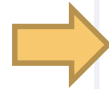
- Напишете програма, която:
  - Получава две **числа**
  - Извежда **"Greater number: "**
  - Отпечатва на конзолата **по-голямото** от тях
- Пример:

5  
8



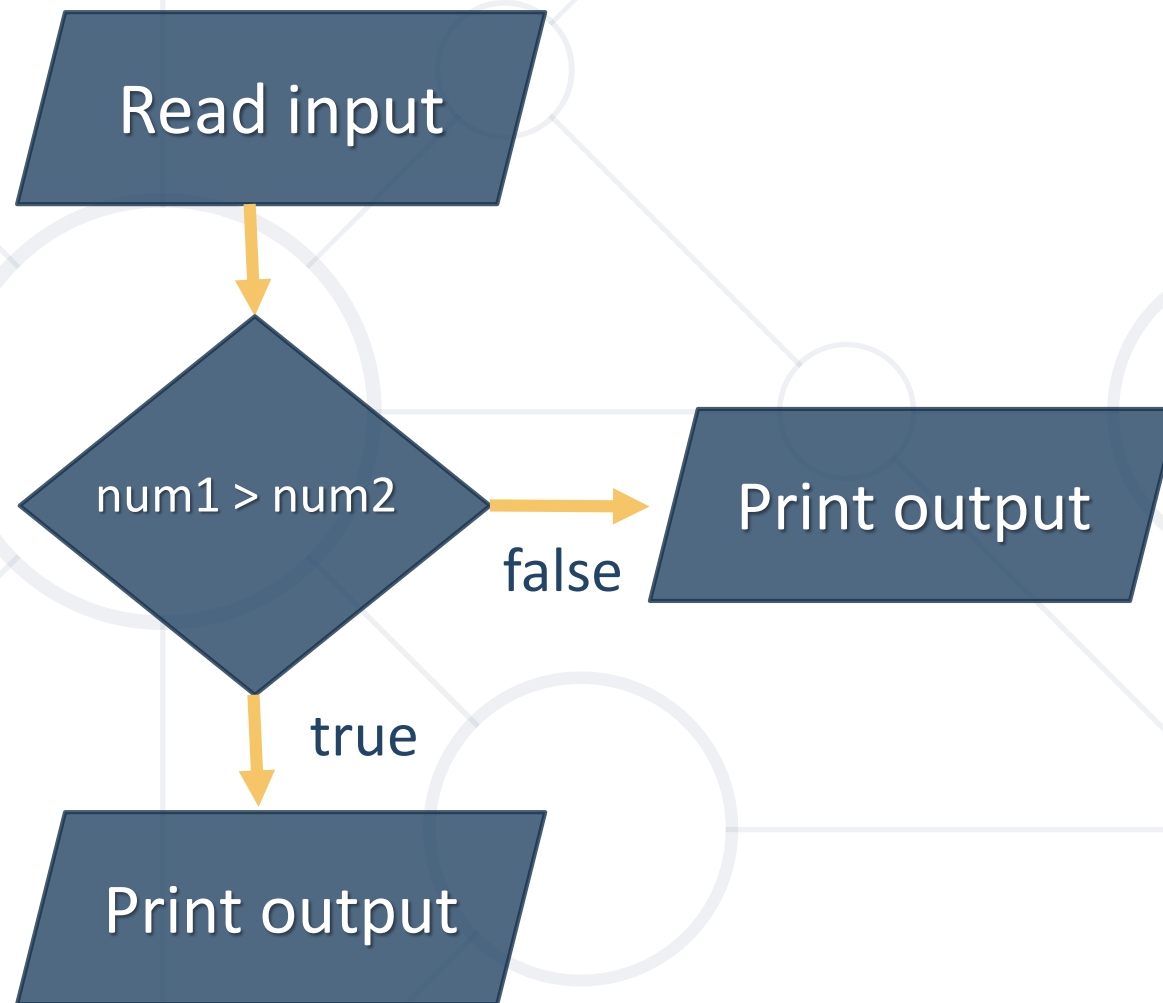
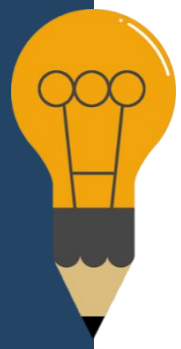
8

7  
3



7





Тестване на решението: <https://judge.softuni.bg/Contests/Compete/Index/1012#2>

# Четно или нечетно число – условие

- Напишете програма, която:
  - Проверява дали едно число е **четно** или **нечетно**
  - Ако е четно извежда **"even"**
  - Ако е нечетно извежда **"odd"**
- Пример:

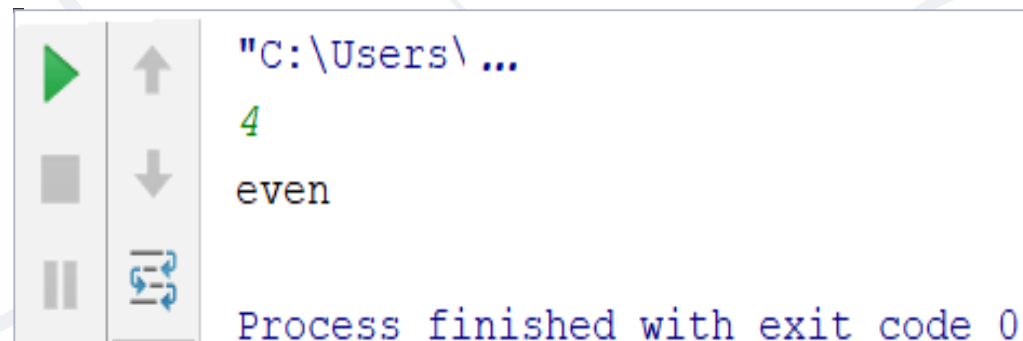
4 → even

7 → odd

4 7

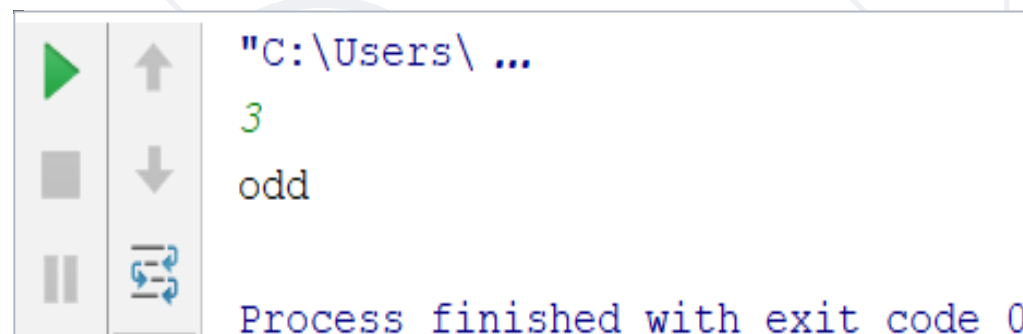
# Четно или нечетно – решение

```
num = int(input())  
if num % 2 == 0:  
    print('even')  
else:  
    print('odd')
```



Terminal output for input 4:

```
"C:\Users\ ...  
4  
even  
Process finished with exit code 0
```



Terminal output for input 3:

```
"C:\Users\ ...  
3  
odd  
Process finished with exit code 0
```




**Серии от проверки**

**По-сложни условни конструкции**



# Серии от проверки

- Конструкцията **if/else-if/else...** може да е в серия




```
if ....:
    # code
elif ....:
    # code
elif ....:
    # code
else:
    # code
```

**TRUE OR FALSE?**

- При истинност на едно условие, **не се продължава** към проверяване на следващите

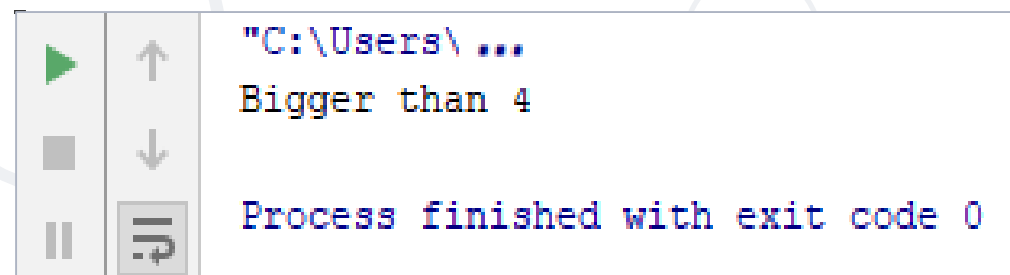
# Серия от проверки - пример

- Програмата проверява първото условие, установява, че е вярно и приключва



```
a = 7
if a > 4:
    print('Bigger than 4')
elif a > 5:
    print('Bigger than 5')
else:
    print('Equal to 7')
```

Извежда се само  
"Bigger than 4"



```
"C:\Users\ ...
Bigger than 4

Process finished with exit code 0
```

# Число от 1 до 9 с текст - условие

- Напишете програма, която:
  - Чете **цяло число**, въведено от потребителя
  - Проверява неговата стойност [1,9]
  - Ако числото е по-голямо от 9 извежда "number too big"
  - Извежда стойността **с текст**
- Пример:

7 → seven

10 → number too big

# Число от 1 до 9 с текст - решение

```
num = int(input())
if num == 1:
    print('one')
elif num == 2:
    print('two')
elif num == 3:
    print('three')
# TODO: add more conditions
else:
    print('number too big')
```



**Живот на променлива**

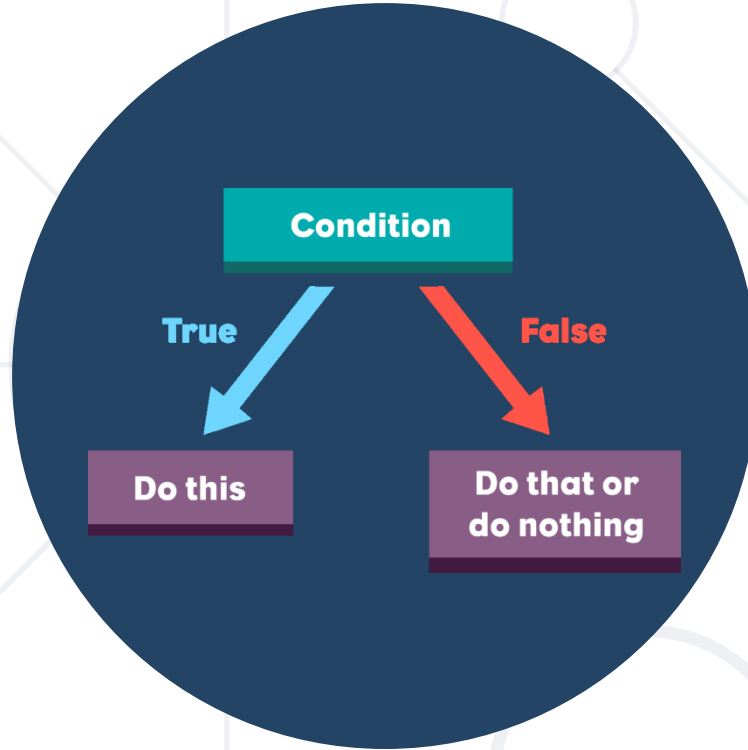
**Диапазон на използване**

- Пример: Променливата **salary** ще съществува **само**, ако е инициализирана някъде в програмата

```
currentDay = "Monday"  
if currentDay == "Monday":  
    salary = 1000  
print(salary)  # 1000
```

- Пример: Променливата **salary** **няма да** съществува, ако не бъде инициализирана някъде в програмата

```
currentDay = "Tuesday"  
if currentDay == "Monday":  
    salary = 1000  
print(salary) # Error
```



# **УСЛОВНИ КОНСТРУКЦИИ**

## **Решаване на задачи в клас (лаб)**



- Напишете програма, която:
  - Прочита **вид** на **геометрична фигура** ("square", "rectangle", "circle" или "triangle")
  - Пресмята **лицето** спрямо вида на фигурата
- Примерен вход и изход:

square  
5



25

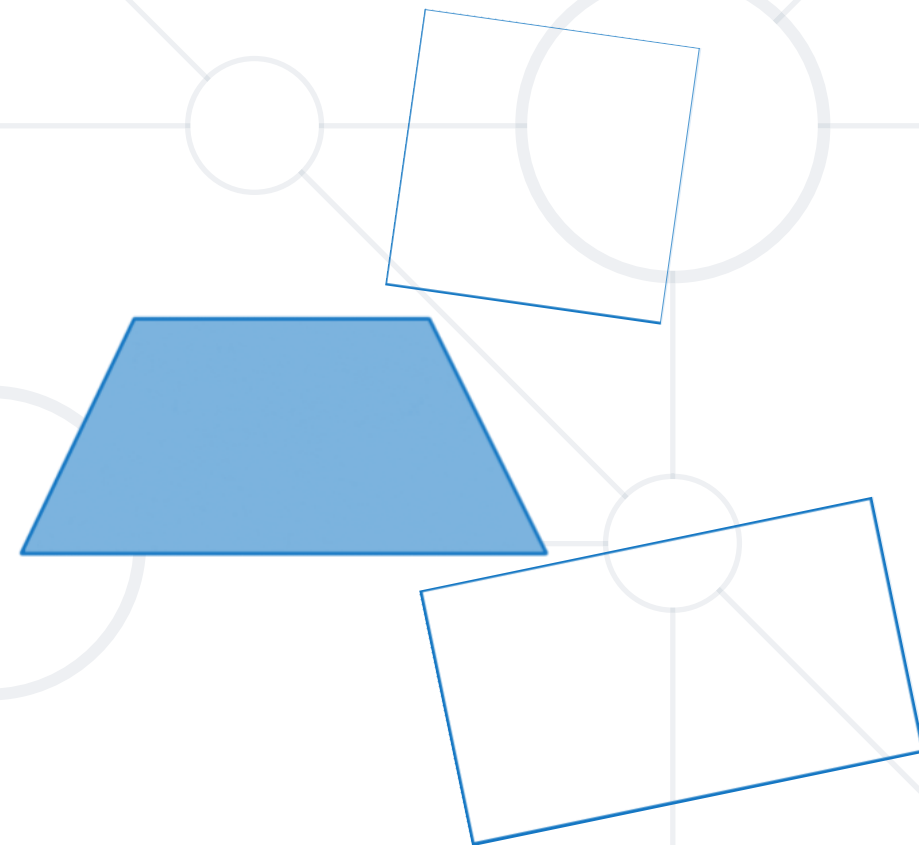
rectangle  
7  
2.5



17.5

# Лица на фигури – решение

```
shape = input()
area = 0.0
if shape == "square":
    side = float(input())
    area = side * side;
elif shape == "rectangle":
    sideA = float(input())
    sideB = input(input())
    area = sideA * sideB;
#TODO: add more conditions
print(area)
```





# Дебъгване

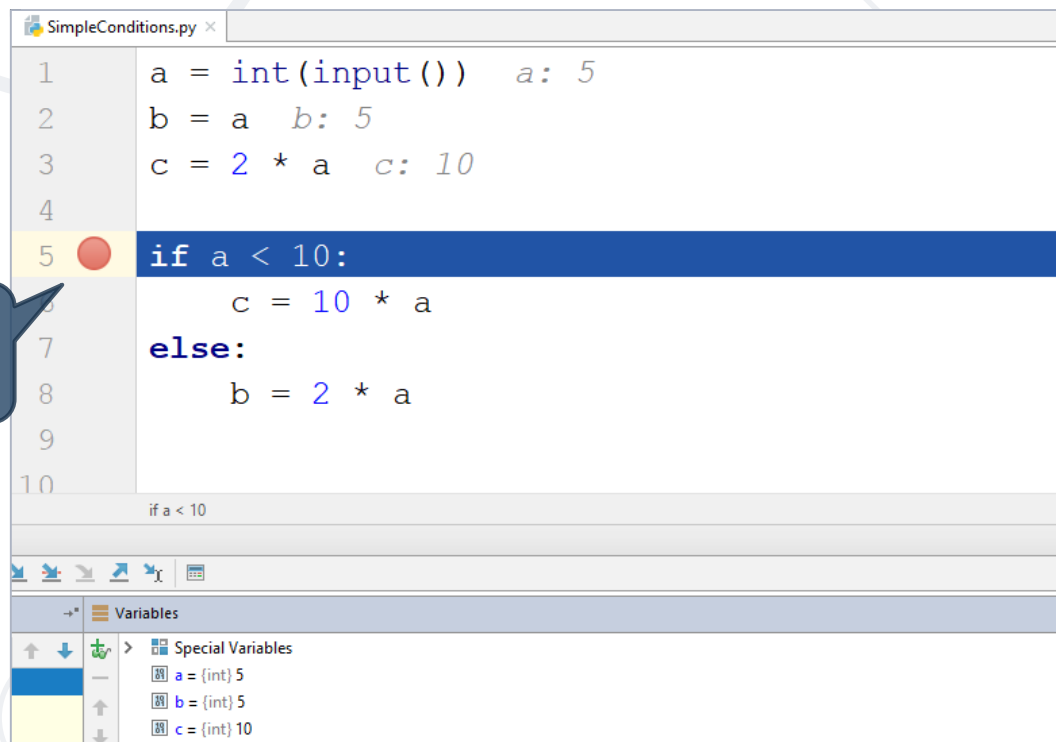
Прости операции с дебъгер

# Дебъгване

- Процес на проследяване на изпълнението на програмата
  - Това ни позволява да откриваме грешки (бъгове)



Breakpoint



```
1 a = int(input()) a: 5
2 b = a b: 5
3 c = 2 * a c: 10
4
5 if a < 10:
6     c = 10 * a
7 else:
8     b = 2 * a
9
10
```

if a < 10

Variables

Special Variables

- a = {int} 5
- b = {int} 5
- c = {int} 10

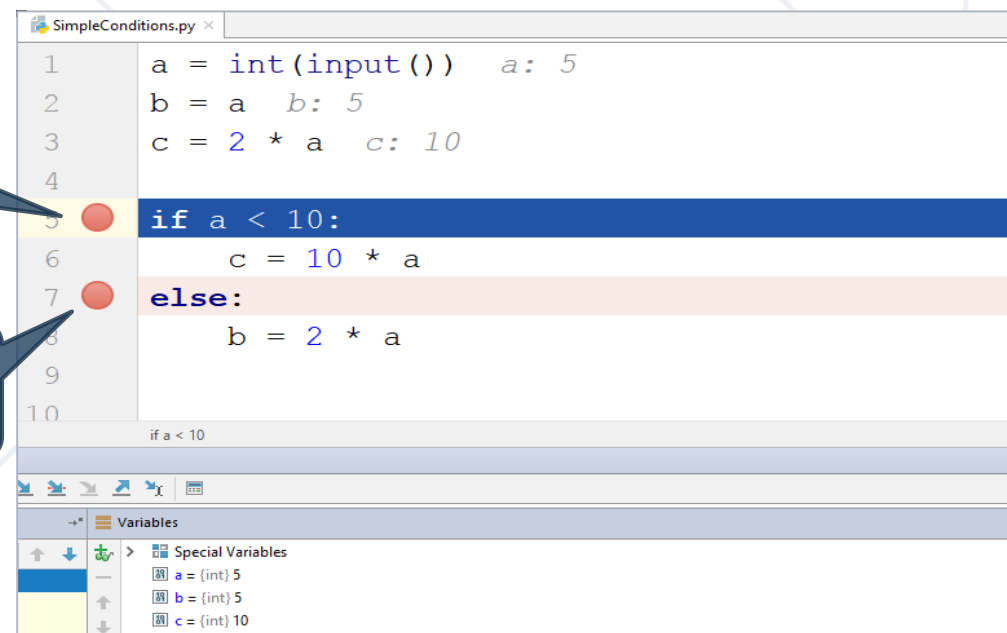
# Дебъгване във PyCharm

- Натискане на [Shift + F9] ще стартира програмата в debug режим
- Можем да преминем към следващата стъпка с [F8]
- Можем да създаваме [Ctrl + F8] стопери – breakpoints
  - До тях можем директно да стигнем използвайки [F9]



Breakpoint

Breakpoint



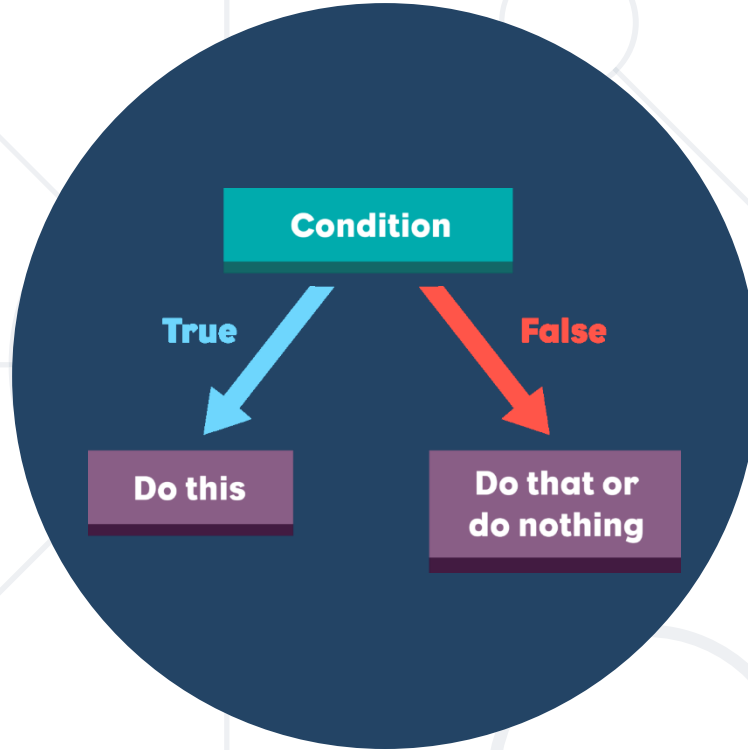
```
SimpleConditions.py x
1 a = int(input()) a: 5
2 b = a b: 5
3 c = 2 * a c: 10
4
5 if a < 10:
6     c = 10 * a
7 else:
8     b = 2 * a
9
10
```

if a < 10

Variables

Special Variables

- a = (int) 5
- b = (int) 5
- c = (int) 10



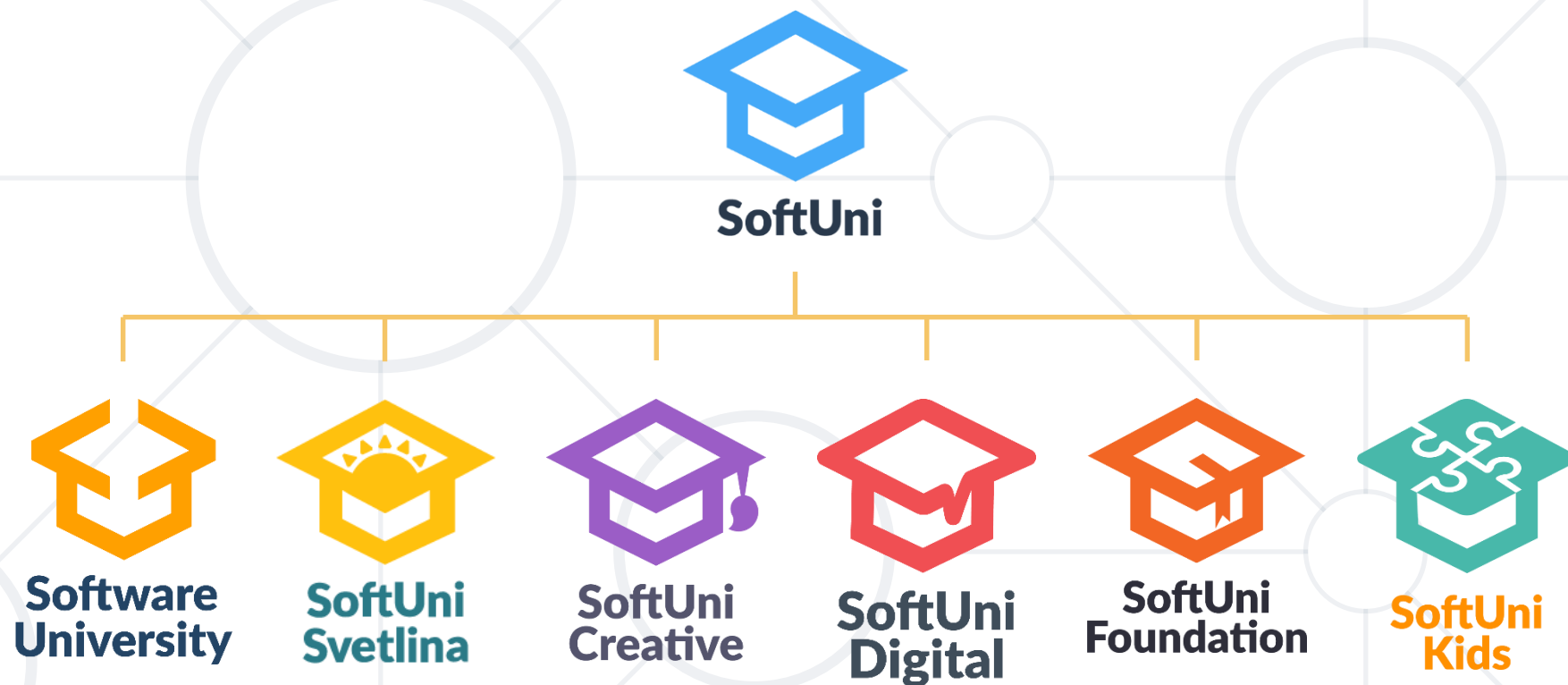
# УСЛОВНИ КОНСТРУКЦИИ

## Решаване на задачи в клас (лаб)

- Конструкции за проверка на условие – **if** и **if-else**
- Живот на променливата
- Дебъгване



# Въпроси?





# SoftUni Diamond Partners



**XS**software



**SBTech**  
*we know sports*



telenor



**SoftwareGroup**  
*doing it right*

**NETPEAK**



**SmartIT**



**Postbank**

*Решения за твоето утре*

**SUPER  
HOSTING  
.BG**

**INDEAVR**

*Serving the high achievers*



**INFRAGISTICS®**



# SoftUni Organizational Partners

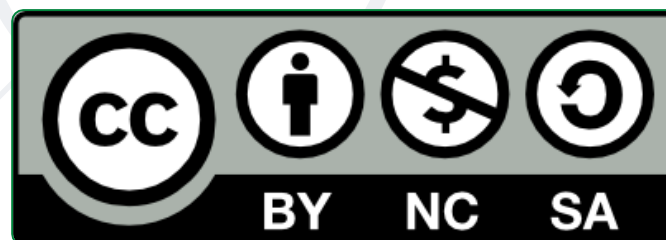


OneBit  
SOFTWARE



WORLD  
OF  
MYTHS

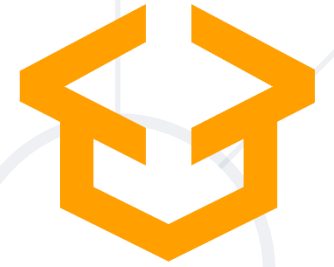
- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



- Благодарности: настоящият материал може да съдържа части от следните източници
  - Книга "Основи на програмирането с Python" от Светлин Наков и колектив с лиценз CC-BY-SA

# Обучения в СофтУни

- Software University – High-Quality Education, Profession and Job for Software Developers
  - [softuni.bg](http://softuni.bg)
- Software University Foundation
  - <http://softuni.foundation/>
- Software University @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)
- Software University Forums
  - [forum.softuni.bg](http://forum.softuni.bg)



**Software  
University**

