

School Nova Computer Science



Handling errors and “while” loop.
Control flow. Types of iterations. While versus For.

Class #4

By Oleg Smirnov

print() and f-strings



f-strings were first implemented in Python 3.6

```
name, age = "Sonya", 12
```

```
print(f"Meet {name}. She is {age} years old.")
```

```
print(f"Meet {name}. "
```

```
    f"She is {age} years old.")
```

```
print(f"Meet {name}. \nShe is {age} years old.")
```

Getting information from user



```
name = input("What's your name? ")  
age = input("What is your age? ")
```

Notice that *age* is a string! You can convert it to an integer data type:

```
age = int(input("What is your age? "))  
print(f"A person half your age would be {age/2} years old")
```

try:



Run this code

except:



Execute this code when
there is an exception

else:



No exceptions? Run this
code.

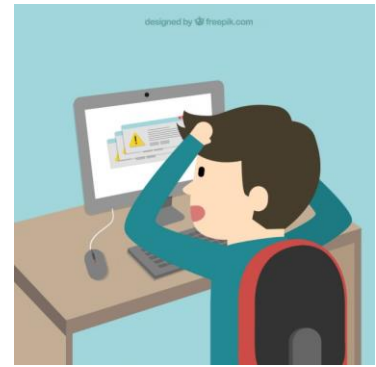
finally:



Always run this code.



Detecting errors



This code identifies if the input was incorrect (not an integer)

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

```
try:
```

```
    age = int(age)
```

```
except:
```

```
    print("Incorrect input")
```

```
else:
```

```
    print(f"A person half your age would be {age/2} years old")
```

```
finally:
```

```
    print("Have a good day!")
```

Handling errors



This code asks the user for an input until an integer is entered

while True:

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

try:

 age = int(age)

except:

 print("Incorrect input")

continue

else:

break

print(f"A person half your age would be {age/2} years old")

Alternative code



This code asks the user for an input until an integer is entered

```
age = 0
```

```
while age == 0:
```

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

```
    try:
```

```
        age = int(age)
```

```
    except:
```

```
        print("Incorrect input")
```

```
        age = 0
```

```
print(f"A person half your age would be {age/2} years old")
```



Iterations: Indefinite loops

Iteration is repeating the same block code. In Python, such iteration is known as a “loop”.

The first kind of iteration:

Indefinite – the loop is repeated until a condition is met (if the condition is never met, the loop will continue forever!) or **break** command is used.

```
current_age, retirement_age = 30, 65
while current_age < retirement_age:
    print(f"You are {current_age} years old. It's too early to retire")
    current_age = current_age + 1
print(f"You reached the retirement age of {retirement_age} years.")
```

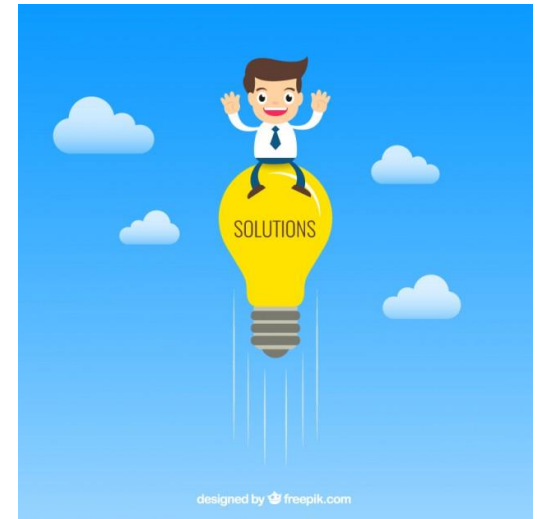

While loop: Example



Write a short script using the while loop, which asks the user to guess a number between 1 and 5. The user has to continue guessing until a correct number is entered.

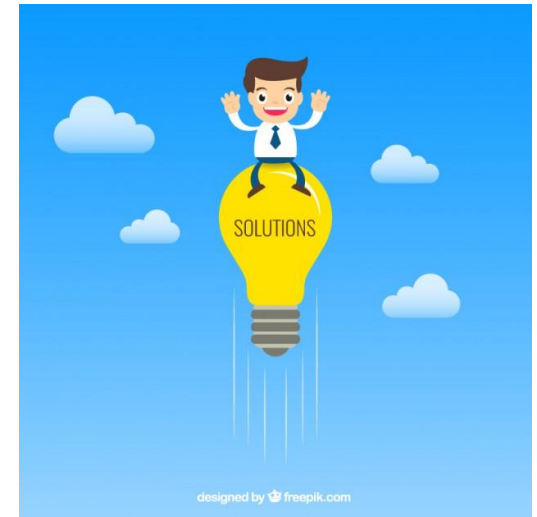
Add a condition: The user has only three attempts to provide a correct answer.

While loop: Example Solution (part 1)



```
correct = 3
guess = 0
while guess != correct:
    guess = int(input("Guess a number between 1 and 5: "))
print(f"Correct! You entered {guess} and the correct answer is {correct}")
```

While loop: Example Solution (part 2)

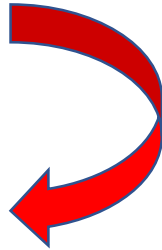


```
correct = 3
guess = 0
attempt = 0
while guess != correct and attempt < 3:
    guess = int(input("Guess a number between 1 and 5: "))
    attempt = attempt + 1
print(f"You entered {guess} and the correct answer is {correct}")
```

While loop: Break and Continue



```
while <condition>:  
    statement 1  
    statement 2  
    break  
    statement 3  
    statement 4  
statement 5
```



```
while <condition>:  
    statement 1  
    statement 2  
    continue  
    statement 3  
    statement 4  
statement 5
```



While loop: Break and Continue Example



```
a = 0
while a < 5:
    a = a + 1
    print("You will see this...")
    break
    print("You will never see this...")
```

```
>>>
You will see this...
```

```
a = 0
while a < 5:
    a = a + 1
    print("You will see this...")
    continue
    print("You will never see this...")
```

```
>>>
You will see this...
You will see this...
You will see this...
You will see this...
You will see this...
```