School Nova Computer Science



Handling errors and "while" loop. Control flow. Types of iterations.

> 10/20/2019 By Oleg Smirnov

Homework comments (part 1)



Common mistake: "=" is not the same as "=="

- a = 5 # assignment, creates a variable *a* equal to 5
- a == 5 # comparison, checks if *a* is equal to 5 and returns **True** or **False**

Which code is better?

a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print(f"Sum of {a} and {b} is {a + b}")
print(f"Product of {a} and {b} is {a * b}")

a = input("Enter first number: ")
b = input("Enter second number: ")
print(f"Sum of {int(a)} and {int(b)} is {int(a) + int(b)}")
print(f"Product of {int(a)} and {int(b)} is {int(a) * int(b)}")

Homework comments (part 2)



Which code is better? (more commands are justified)

a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print(f"Sum of {a} and {b} is {a + b}")
print(f"Product of {a} and {b} is {a * b}")

```
a = input("Enter first number: ")
b = input("Enter second number: ")
try:
```

```
a, b = int(a), int(b)
except:
```

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

```
else:
```

```
print(f"Sum of {a} and {b} is {a + b}")
print(f"Product of {a} and {b} is {a * b}")
```

Homework: 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")

Homework: 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.")</pre>
```

While loop: Classroom exercise



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: Classroom exercise 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}")

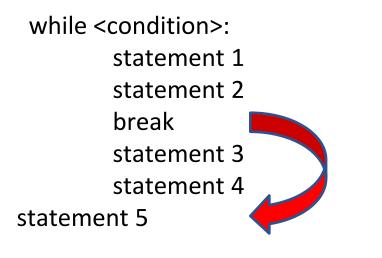
While loop: Classroom exercise Solution (part 2)

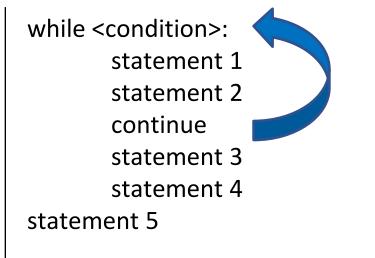


```
correct = 3
guess = 0
attempt = 0
while guess != correct and attempt < 3:
    guess = int(input("Guess a number between 1 and 5: "))
    attempt = attempt + 1
print(f"You entered {guess} and the correct answer is {correct}")</pre>
```

While loop: Break and Continue







While loop: Break and Continue Example



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