

```
{:title "Python - If-else-statements" :layout :post :date "2016-10-09" :tags
["Python" "Code" "Guide"]}
```

Previously you learned about booleans and how to do logical statements. Now we'll learn how to use the booleans to do conditional statements in Python. Conditional statements like in many programming languages call them if-else-statements. If this is true then this is done else this other thing is done. You'll see soon enough what I mean. If-Else-Statements take a condition and if the condition is true will do the expression otherwise it will do the other expression since it is false. Enough of me just talking see for yourself how it works. Open up a text editor and type in the following if-else statements.

```
def main():

    if (True):
        print("I am printing because I am True!")

    if (False):
        print("I am not going to print because I am False!")

    age = 19

    if (age > 10):
        print("I am older than 10 years.")

    if (age > 20):
        print("I am older than 20 years.")
    else:
        print("I am younger than 20 years.")

    grade = 81

    if (grade > 90):
        print("I got over 90.")
    elif (grade > 80):
        print("I got over 80")
    elif (grade > 70):
        print("I got over 70.")
    else:
        print("I got under 70.")

    if (grade > 90 and age > 20):
        print("I got over a 90 and I am over 20 years old.")
```

```

elif (grade > 70 and age > 20):
    print("I got over a 70 and am over 20 years old.")
else:
    print("I got a", grade, "and am over", age, "years old.")

main()

```

Result:

```

I am printing because I am True!
I am older than 10 years.
I am younger than 20 years.
I got over 80
I got a 81 and am over 19 years old.

```

Explanation below: Line 3-7:

```

if (True):
    print("I am printing because I am True!")

if (False):
    print("I am not going to print because I am False!")

```

These are if statements. Just testing for if the condition which is in parantheses are TRUE. This is why the second if statement did not print in the result and only the one TRUE did. The basic format for writing a if statement is starting with the if keyword in python which starts the if statement. Followed by the parantheses which you put the condition you are testing in this case it's already a Boolean. Followed by a : to end the line. Than you indent the expression you want if that condition is met to be evaluated underneath. Line 9-17:

```

age = 19

if (age > 10):
    print("I am older than 10 years.")

if (age > 20):
    print("I am older than 20 years.")
else:
    print("I am younger than 20 years.")

```

On line 10 we have our age as a variable set to 19. On line 11 we have a if statement checking if our age is greater than 10. It is so it prints. Line 14 we have a if-else statement which is similiar to an if statement expect it has a

default value if none of the conditions hold true. In this cause age is not greater than 20 and so it prints what is under the else statement. The else: will come after the if when writing if-else statements. Else does not take any conditions if you do it would be an error. Line 19-28:

```
grade = 81

if (grade > 90):
    print("I got over 90.")
elif (grade > 80):
    print("I got over 80")
elif (grade > 70):
    print("I got over 70.")
else:
    print("I got under 70.")
```

What if you wanted to test for multiple conditions? Well then you write a if-elif-else condition. Elif works in a similar way that if does that it takes a condition. However it elif follows after a if statement. One thing to notice is that none of the other prints under the elif (grade > 80) are even printed or even checked for. Once one of the conditions are met to be true that it leaves the if statement and directly into the body of the elif and does whatever is in the body. Line 30-35:

```
if (grade > 90 and age > 20):
    print("I got over a 90 and I am over 20 years old.")
elif (grade > 70 and age > 20):
    print("I got over a 70 and am over 20 years old.")
else:
    print("I got a", grade, "and am over", age, "years old.")
```

In this if-elif-else statement we are using and to test that both of the conditions are met so the first one is False and False which if you remember using and if one of them is False the entire condition is evaluated as False. So the if part of the statement is not met. Then the elif statement is True and False since the grade is higher than 70 but the age is not again this is evaluated as False and the elif part is not met. Thus it defaults to the else statement.

Wrapping up: So we learned if statements, if-else statements and if-elif-else-statements. One thing I did leave out here is nested if-else and if-elif-else statements which are a bit more tricky but basically the same principle. One thing to keep in mind is that using if-statements you want to be careful what conditions you are checking since when the first condition is found that holds true that condition's body will be evaluated and nothing else will even be checked. If statements are widely used and are important tools to use in most programming languages as a part of decision making and making choices in programming. So making sure you understand them is critical.