If and else statement

Asking question using if and else statement

In programming, we often ask yes or no questions, and decide to do something based on the answer. For example, we might ask, "Are you older than 21?" and if the answer is yes, respond with "You are an adult!"

Those sorts of questions are called *conditions*, and we combine these conditions and the responses into *if* statement. Conditions can be more complicated than a single question, and *if* statements can also be combined with multiple questions and different responses based on the answer to each questions.

Conditions help us compare things

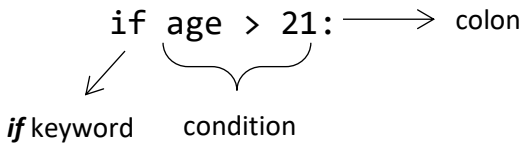
A condition is a programing statement that compares things and tells us whether the criteria set by the comparison are either True of False.

We use symbols is Python, called *comparison operators*, to create our conditions:

Python Comparison Operators	
Symbol	Definition
`==`	Equal to
`!=`	Not equal to
`>`	Greater than
`<`	Less than
`>=`	Greater than or equal to
`<=`	Less than or equal to

if statement

An *if* statement is a conditional statement that check if the statement is true. The *if* statement in Python is made up of the *if* keyword, followed by a condition and a colon:



After the colon, the next line should be command blocks, which will run if the conditional is True, otherwise, the program will skip the entire command blocks. Python identifies command blocks with a tab (inserted when you press the TAB key) or four spaces at the beginning from left to right:

```
if age>= 21:
    print("You are an adult!")
```
← Command block

For example, if we assign value 20 to variable *age* in the following program:

```
*example1.py - C:\Users\Student\Desktop\pythonFile\example1.py (3.6.5)*
File  Edit  Format  Run  Options  Window  Help
age = 20;
if age>= 21:
    print("You are an adult!")
    print("This paragraph is inside the if statement")
print("This paragraph is outside the if statement")
                                                    Ln: 1  Col: 8
```

Since the statement is False, what is inside the *if* statement will not display. Then it will run only the line after the *if* statement

```
Python 3.6.5 Shell
File  Edit  Shell  Debug  Options  Window  Help
========== RESTART: C:\Users\Student\Desktop\pythonFile\example1.py ==========
This paragraph is outside the if statement
>>>
                                                    Ln: 98  Col: 4
```

If we assign the value 25 to variable *age*, the statement will be true, the command blocks inside the *if* statement will run, and then the lines after the *if* statement.

```
*example1.py - C:\Users\Student\Desktop\pythonFile\example1.py (3.6.5)*
File  Edit  Format  Run  Options  Window  Help
age = 25;
if age>= 21:
    print("You are an adult!")
    print("This paragraph is inside the if statement")
print("This paragraph is outside the if statement")
                                                    Ln: 2  Col: 8
```

```
Python 3.6.5 Shell
File  Edit  Shell  Debug  Options  Window  Help
>>>
========== RESTART: C:\Users\Student\Desktop\pythonFile\example1.py ==========
You are an adult!
This paragraph is inside the if statement
This paragraph is outside the if statement
>>>
                                                    Ln: 103  Col: 4
```

In Python, *whitespace*, is meaningful. Code that is at the same position (indented the same number of spaces from the left margin) is grouped into a block, and whenever you start a new line with more spaces than the previous one:

```
Block 1 = line of code2
      Block 1 = line of code3
            Block 2 = line of code4 (inside of code3)
            Block 2 = line of code5 (inside of code3)
                  Block 3 = line of code6 (inside of code5)
            Block 2 = line of code7 (inside of code3)
      Block 1 = line of code8

            Block 4 = line of code9 (inside of code8)
```

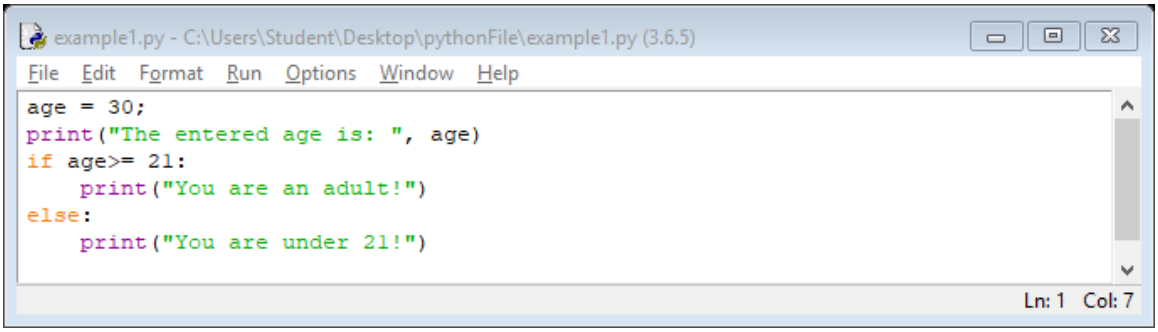
We group statements together into blocks because they are related. The statements need to be run together. When we change the indentation, we are generally creating new blocks.

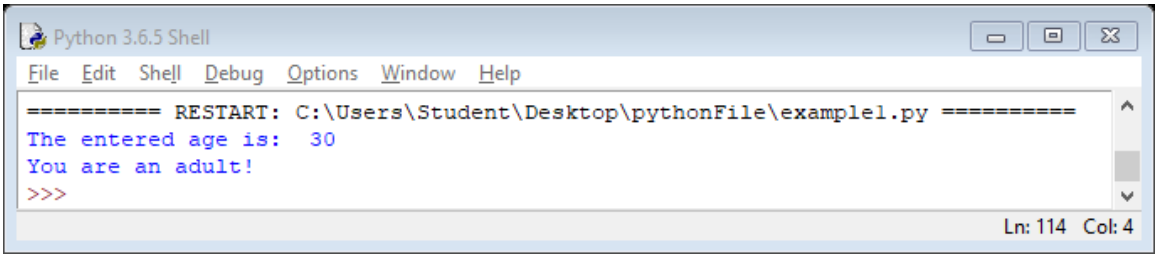
From the previous blocks of line, even though Block 2 and Block 4 have the same indentation, they are considered different blocks because Block 2 is inside of code 3 and Block 4 is inside of code 8.

If – else statement

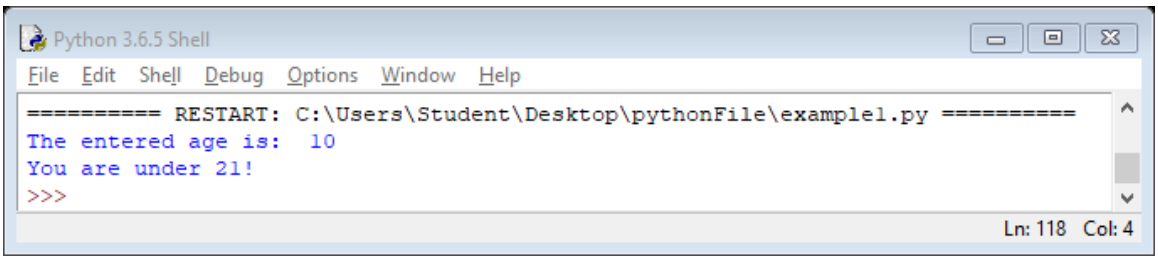
In addition to use *if* statements to do something when a condition is True, we can also use *if* statements to do something when a condition is not true. For example, we might print one message to the screen if your age is 21 (True) and another if your age is not 21 (False). The trick here is to use an *if-else* statement, which essentially says "if the input age is greater than or equal to 21, then print: You are an adult!; or else, print: You are under 21!"

For example, if we enter age = 30, then *if* statement is true, therefore it will run the block inside the *if* statement:





Otherwise, if you enter age = 10, then *if* statement is false, therefore it will run the block inside the *else* statement:



if-elif-else statement

We can extend an *if* statement even further with *elif*, which is a short for else-if. For example, we can check if a person's age is 10, 11, 12, and so on.

Example) check the age between 5 and 10. For each age, print their average height and weight using the following table:

Age	Height	Weight
5	102	14.8
6	108	16.3
7	113	18.0
8	119	19.7
9	123	21.5
10	124	23.5

If the age is not between 5 and 10, then print: *Unable to display your height and weight*. For example, if we set the age value to 10:

example1.py - C:\Users\Student\Desktop\pythonFile\example1.py (3.6.5)

File Edit Format Run Options Window Help

```
age = 10;
if age == 5:
    print("Your age is 5 years old")
    print("Your height should be: 102 cm and your weight should be: 14.8 lbs")
elif age ==6:
    print("Your age is 6 years old")
    print("Your height should be: 108 cm and your weight should be: 16.3 lbs")
elif age ==7:
    print("Your age is 7 years old")
    print("Your height should be: 113 cm and your weight should be: 18.0 lbs")
elif age ==8:
    print("Your age is 8 years old")
    print("Your height should be: 119 cm and your weight should be: 19.7 lbs")
elif age ==9:
    print("Your age is 9 years old")
    print("Your height should be: 123 cm and your weight should be: 21.5 lbs")
elif age ==10:
    print("Your age is 10 years old")
    print("Your height should be: 124 cm and your weight should be: 23.5 lbs")
else:
    print("Your age is ", age, " years old")
    print("Unable to display your height and weight")
```

Ln: 1 Col: 7

Python 3.6.5 Shell

File Edit Shell Debug Options Window Help

```
========== RESTART: C:\Users\Student\Desktop\pythonFile\example1.py ==========
Your age is 10 years old
Your height should be: 124 cm and your weight should be: 23.5 lbs
>>>
```

Ln: 122 Col: 4

If we set age = 20:

Python 3.6.5 Shell

File Edit Shell Debug Options Window Help

```
========== RESTART: C:\Users\Student\Desktop\pythonFile\example1.py ==========
Your age is  20  years old
Unable to display your height and weight
>>>
```

Ln: 129 Col: 4

Combining Conditions

We can combine conditions by using the keywords ***and*** and ***or*** logical operator.

Operator	Description	Example
and	Returns TRUE if both statements are TRUE	age > 5 **and** age <10
or	Returns TRUE if one of the statements is TRUE	age == 5 **or** age == 6
not	Reserve the result, returns FALSE if the result is TRUE	**not**(age > 5 **and** age <10)

The and operator

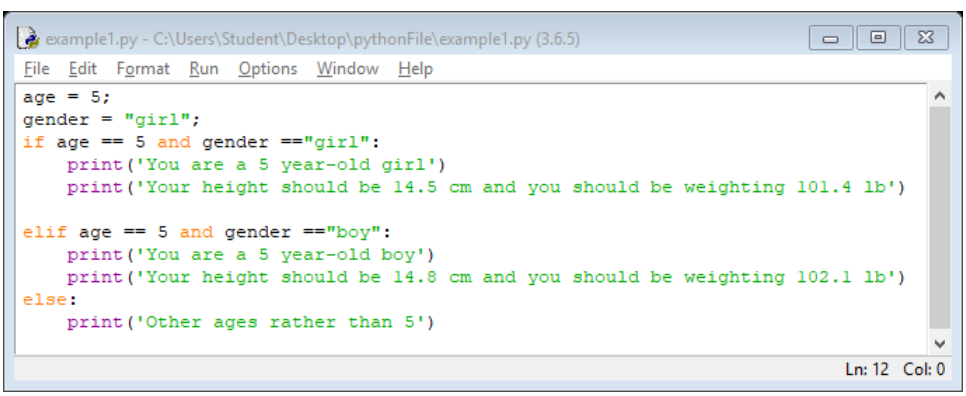
The *and* operator is used to indicate that the statement will be true when all the conditions are true. For example, if we want to print that if the kid is a 5 year-old girl, her weight and height should 14.5cm and 101.4 lb respectively:

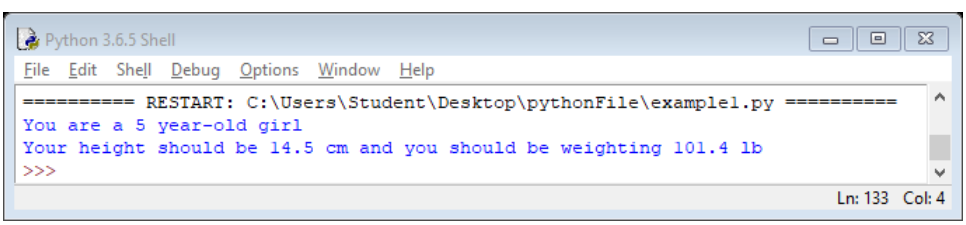
```
if age == 5 and gender =="girl" :
```

We can also add to the code that if the kid as a 5 year-old boy, his weight and height should 14.8 cm and 102.1 lb respectively:

```
elif age == 5 and gender =="boy":
```

The complete code will be as the following:





The or operator

The ***or*** operator is used to indicate that the statement will be true when at least one of the condition is true. For example, if a kid is between the age of 5 and 6, it will print: *`You should eat for pasta and veggies than candies!`*

```
if age ==5 or age == 6:
    print("You should eat more pasta and cheese than candies!")
```

We can create a code using logical conditions as following:

Ages	Diet message
5 and 6	You should eat more pasta and cheese than candies
7 and 8	It is time to replace donuts with fisher and veggies
9 and 10	Welcome to milkshake with fruit and grains

The complete code will look like: