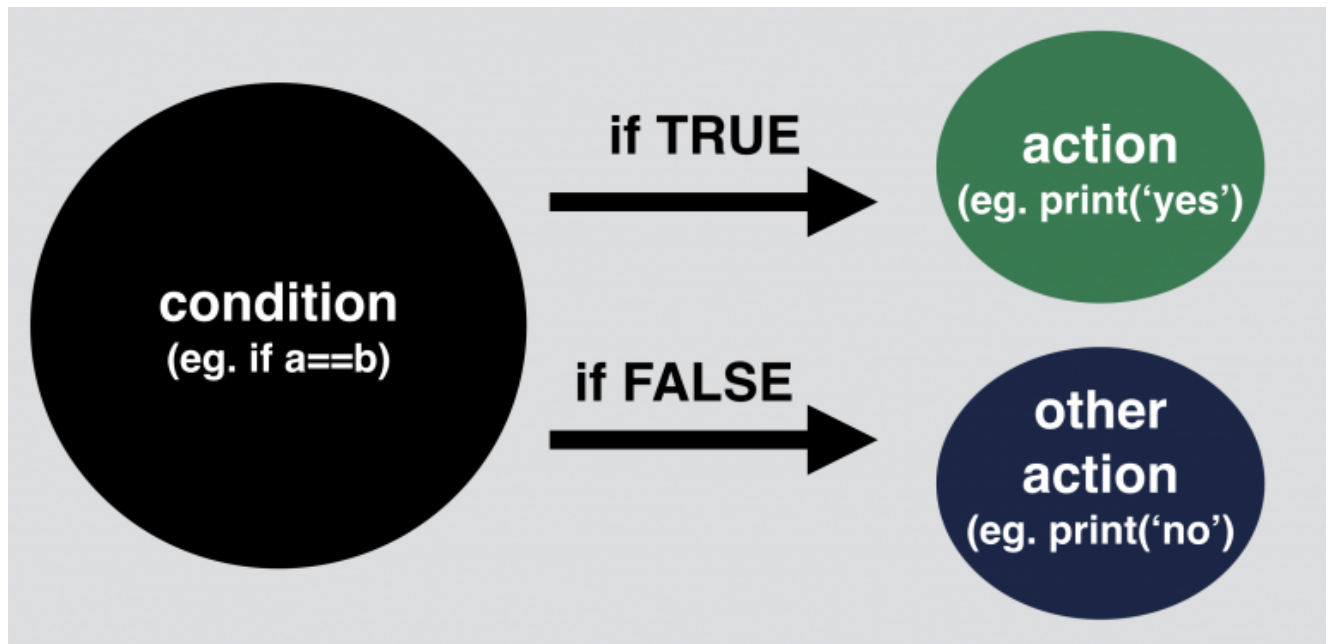


If-Else

Introduction

We use if statements in our everyday life all the time - even if our everyday life is not written in Python. If the light is green then I'll cross the road; otherwise I'll wait. If the sun is up then I'll get out of bed; otherwise I'll go back to sleep. Okay, maybe it's not this direct, but when we take actions based on conditions, our brain does what a computer would do: evaluate the conditions and act upon the results. Well, a computer script doesn't have a subconscious mind, so for practicing data science we have to understand how an if statement works and how we can apply it in Python!



Let's say we have two values: $a = 10$ and $b = 20$. We compare these two values: $a == b$. This comparison has either a True or a False output. (Test it in your Jupyter Notebook!)

```
In [1]: a = 10  
        b = 20  
        a == b
```

```
Out[1]: False
```

We can go even further and set a condition: if $a == b$ is True then we print 'yes'. If it's False then we print 'no'. And that's it, this is the logic of the Python if statements. Here's the syntax:

```
In [1]: a = 10
        b = 20
        if a == b:
            print('yes')
        else:
            print('no')
```

no

Run this mini script in your Jupyter Notebook! The result will be (obviously): no.

Now, try the same - but set b to 10!

The returned message is yes.

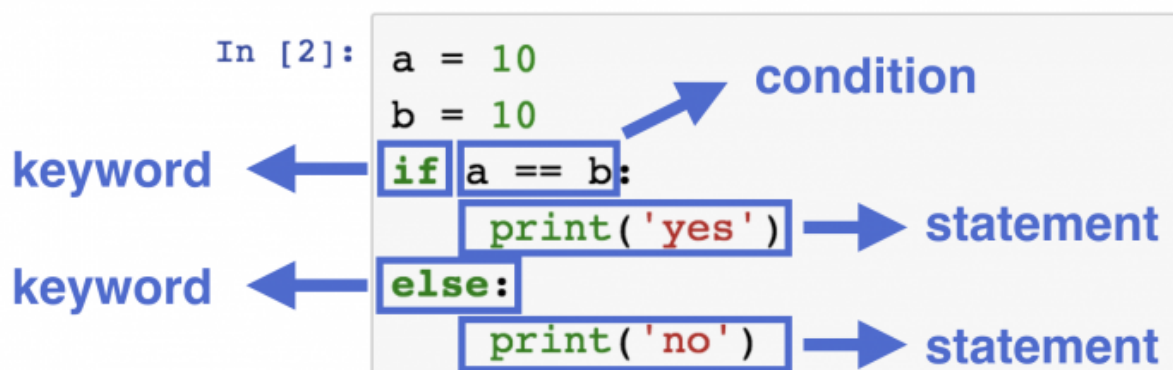
```
In [2]: a = 10
        b = 10
        if a == b:
            print('yes')
        else:
            print('no')
```

yes

Python if statement syntax

Let's take a look at the syntax,
because it has pretty strict rules.

The basics are simple:



You have:

1. an if keyword, then
2. a condition, then
3. a statement, then

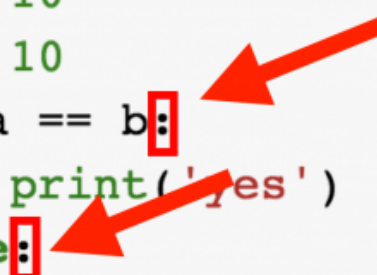
4.an else keyword, then

5.another statement.

6.However, there are two things to watch out for:


1. Never miss the colons at the end of the if and else lines!

```
In [2]: a = 10
        b = 10
        if a == b:
            print('yes')
        else:
            print('no')
```



2. And never miss the indentation at the beginning of the statement-lines!

```
In [2]: a = 10
        b = 10
        if a == b:
            print('yes')
        else:
            print('no')
```



If you miss any of the above two, an error message will be returned saying "invalid syntax" and your Python script will fail.

Python if statements - level 2

Now that you understand the basics, it's time to make your conditions more complex - by using arithmetic, comparison and logical operators. (Note: if the word "operators" does not ring any bells, you might want to check out this article first:

This script will return yes, since both of the conditions, $(a + b) / c == 1$ and $c - b - a == 0$ are actually True and the logical operator between them was: and.

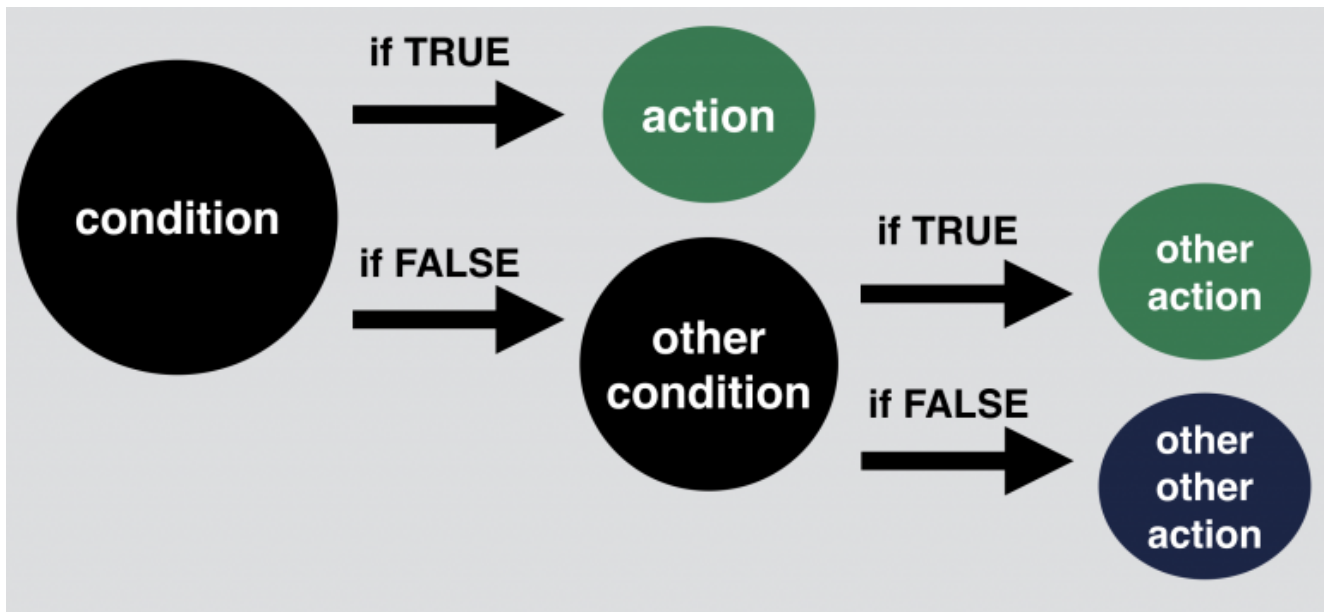
```
In [3]: a = 10
        b = 20
        c = 30
        if (a + b) / c == 1 and c - b - a == 0:
            print('yes')
        else:
            print('no')
```

yes

Of course, you can make this even more complex if you want, but the point is: having multiple operators in an if statement is absolutely possible - in fact, it's pretty common in real life scenarios!

Python if statements - level 3

You can take it to the next level again, by using the elif keyword (which is a short form of the "else if" phrase) to create condition-sequences. "Condition-sequence" sounds fancy but what really happens here is just adding an if statement into an if statement:



Sure enough the result will be "second condition is true".

```
In [4]: a = 10
        b = 11
        c = 10
        if a == b:
            print('first condition is true')
        elif a == c:
            print('second condition is true')
        else:
            print('nothing is true. existence is pain.')
```

second condition is true

You can do this infinite times, and build up a huge if-elif-elif-...-elif-else sequence if you want!
And... This was more or less everything you have to know about Python if statements.