

## Day 4

In day 04, Harry only told you about print statements on how to use them, in short **print()** is a method which takes a valid **object** inside parenthesis. And when I write **print("Hello, World!")** this is known as print statement. Now how print statement works and what's inside that code you seriously don't need to get into that right now because you are just starting up and later on I promise you will be able to build your own functions.

But these notes that I create are only for your plus knowledge so let me just tell you about **sep & end** in **print()** method. You know till now that you can give multiple values inside the **print()** method like,

```
print("Hello world",7)
```

Like that you can give multiple values right, what if between each value or after each value you want to add something like "@" after each value so the answer should become,

```
Hello world@7
```

Well here comes **sep** inside the **print()** method that can help you achieve that. Now what is a **sep**? well, The **sep** parameter in **print()** controls what is placed between the values you print.

```
print("Hello world",7,sep="@")
```

You can pass in **sep**=(a valid value), by doing like this you will get a following output:

```
Hello world@7
```

Now what is this thing called when I am writing **sep**= what it means like how **print()** method is understanding this? You will get to know about it later on but first go and try this, learn how to use it. I promise it will make sense to you after some videos. And also remember one more thing about **sep** is that when we don't pass the value of **sep** like when we write simple print statements for example

```
print("Hello world",7)
```

By-default when we don't specify the value of **sep** it will become **sep= ""** like empty space because when you just print the following statement:

```
print("Hello world",7)
```

you will get a following output:

```
Hello world 7
```

See how it just added a default space after **world**, because remember when we don't specify the value of **sep** its default value will be then **"" an empty space**. I hope you understand this, if not then go and try out this code by making little changes in it and then you will understand about it.

Now just like **sep**= we have something called "**end**", The end parameter in print() controls what comes at the end of the printed line.

For example:

```
print("Hello world",end="@@")
```

The output I will get is this:

```
Hello world@@
```

See, whatever I added as the value of **end** , it just added it at the end of the string. Just like when we don't specify the value of **sep** it adds a empty space by-default , same goes for **end** when we don't specify its value by default, it moves to a new line, but you can change it to something else, like a space or a dot, to control how the output looks.

Another example that explains the **by-default** value of **end** is,

```
print("Hello")  
print("World")
```

for this code the output I will get is,

```
Hello  
World
```

This is because after first print statement by-default a statement ended with a new line that's why we are seeing **world** in a next line and not in the same line.

But lets add a empty space as the value of **end** will I get a output on the same line?

```
print("Hello",end=" ")  
print("World")
```

the output I will get is,

```
Hello World
```

As you can see that im getting a output on the same line because now I specified the value of **end** that its value is empty space not a default value which is a newline.

Now what is newline how it adds a newline this and that I know you are getting questions in your mind but that's what you are going to see in the next video.

Another thing you need to understand is **sep** & **end** only accepts values with in single or double quotes. For now understand this that it accepts values inside single or double quotes that's it.