Python pass

Summary: in this tutorial, you'll learn how to use the Python `pass` statement as a placeholder.

Introduction to the Python pass statement

Suppose that you have the following `if...else` (https://www.pythontutorial.net/python-basics/python-if/) statement:

```
counter = 1
max = 10
if counter <= max:
    counter += 1
else:
    # implement later
```

In the `else` clause, you haven't got any code yet. But you'll write code for this `else` clause later.

In this case, if you run the code, you'll get an syntax error (`SyntaxError`).

This is where the Python `pass` statement comes into play:

```
counter = 1
max = 10
if counter <= max:
    counter += 1
else:
    pass
```

The `pass` statement is a statement that does nothing. It's just a placeholder for the code that you'll write in the future.

When you run the code that contains a `pass` statement, the Python interpreter will treat the `pass` statement as a single statement. As a result, it doesn't issue a syntax error.

Technically, you can use the `pass` statement in many statement in Python.

Let's take some examples of using the `pass` statement.

1) Using the Python pass statement with the if statement example

The following shows how to use the `pass` statement with an `if` statement:

```
if condition:
    pass
```

2) Using the Python pass statement with the for statement

This example shows how to use the `pass` statement in a `for` (https://www.pythontutorial.net/python-basics/python-for-range/) loop:

```
for i in range(1,100):
    pass
```

3) Using the Python pass statement with the while statement

The following example shows how to use the `pass` statement with a `while` (https://www.pythontutorial.net/python-basics/python-while/) loop:

```
while condition:
    pass
```

4) Using the Python pass statement with functions and classes

Later, you'll learn how to define a function (https://www.pythontutorial.net/python-basics/python-functions/) :

```
def fn():
    pass
```

and a class (https://www.pythontutorial.net/python-oop/python-class/) :

```
class Stream:
    pass
```

In these examples, you use the `pass` statement to mark the function and class empty.

Summary

- Use the Python `pass` statement to create a placeholder for the code that you'll implement later.