Python **kwargs

Summary: in this tutorial, you'll learn about the Python `**kwargs` parameters.

Introduction to the Python **kwargs parameters

In Python, a function (https://www.pythontutorial.net/python-basics/python-functions/) can have a parameter preceded by two stars (**). For example: `**kwwargs`

The `**kwargs` is called a keyword parameter.

When a function has the `**kwargs` parameter, it can accept a variable number of keyword arguments (https://www.pythontutorial.net/python-basics/python-keyword-arguments/) as a dictionary (https://www.pythontutorial.net/python-basics/python-dictionary/) .

The two stars (`**`) are important. However, the name `kwargs` is by convention. Therefore, you can use any other meaningful names such as `**configs` and `**files` .

The following example defines a function called `connect()` that accepts a `**kwargs` parameter:

```
def connect(**kwargs):
    print(type(kwargs))
    print(kwargs)
```

The following function call shows an empty dictionary to the screen:

```
connect()
```

Output:

```
<class 'dict'>
{}
```

In this example, we didn't pass any arguments to the `connect()` function, the `kwargs` is empty dictionary.

The following calls the `connect()` function and passes some keyword arguments into it:

```
connect(server='localhost', port=3306, user='root', password='Py1hon!Xt')
```

It shows the following dictionary to the screen:

```
<class 'dict'>
{'server': 'localhost', 'port': 3306, 'user': 'root', 'password': 'Py1hon!Xt'}
```

Inside the `connect()` function, you can use the `kwargs` argument as a dictionary.

If you want to pass a dictionary to the function, you need to add two stars (`**`) to the argument like this:

```
def connect(**kwargs):
    print(kwargs)


config = {'server': 'localhost',
        'port': 3306,
        'user': 'root',
        'password': 'Py1thon!Xt12'}

connect(**config)
```

If a function has the `**kwargs` parameter and other parameters, you need to place the `**kwargs` after other parameters. Otherwise, you'll get an error.

The syntax of the following `connect()` function is correct:

```
def connect(fn, **kwargs):
    print(kwargs)
```

However, the syntax of this function causes a SyntaxError:

```
def connect(**kwargs, fn):
    print(kwargs)
```

Using both *args and **kwargs arguments

The following function has both *args and **kwargs parameters:

```
def fn(*args, **kwargs):
    print(args)
    print(kwargs)
```

The `fn` function can accept a variable number of the positional arguments. Python will pack them as a tuple and assign the tuple to the `args` argument.

The `fn` function also accepts a variable number of keyword arguments. Python will pack them as a dictionary and assign the dictionary to the `kwargs` argument.

For example:

```
fn(1, 2, x=10, y=20)
```

Output:

```
(1, 2)
{'x': 10, 'y': 20}
```

Summary

- Use the Python `**kwargs` parameter to allow the function to accept a variable number of keyword arguments.
- Inside the function, the `kwargs` argument is a dictionary that contains all keyword arguments as its name-value pairs.
- Precede double stars (`**`) to a dictionary argument to pass it to `**kwargs` parameter.
- Always place the `**kwargs` parameter at the end of the parameter list, or you'll get an error.