Python regex `group` functions

Python Regex group() function explained with examples: named groups and groupdict.

We'll cover the following

- Group dictionary `Groupdict`

A regular expression can have **named** groups. This makes it easier to retrieve those groups after calling `match()`. But it makes the pattern more complex.

Following example shows a named group (`first` and `last`).

```
#!/usr/bin/python
import re

# A string.
name = "Learn Scientific"

# Match with named groups.
m = re.match("(?P<first>\w+)\W

# Print groups using names as
if m:
    print(m.group("first"))
    print(m.group("last"))
```

We can get the first name with the string **"first"** and the `group()` method. We use **"last"** for the last name.

Group dictionary `Groupdict`

A regular expression with named groups can fill a dictionary. This is done with the `groupdict()` method. In the dictionary, each group name is a `key` and

Each value is the data matched by the regular expression. So we receive a key-value store based on groups.

```
import re

name = "Scientific Python"

# Match names.
m = re.match("(?P<first>\w+)\W+(?P<last>\w+)", name)

if m:
    # Get dict.
    d = m.groupdict()

    # Loop over dictionary with for-loop.
    for t in d:
        print("  key:", t)
        print("value:", d[t])
```