Go Back to Pre Work

Course Content

1.3 Good practices of naming variables - Summary Notes

A variable name should be

a. easy to code - The variable name should be easy to code. For example, among the variable names "weight_of_a_patient", "weightofapatient", and "weight", the variable name "weight" looks more clear, concise, and easy to code as it may be used multiple times in the program and it might become tedious to write lengthy names (like "weight_of_a_patient" and "weightofapatient") every time.

b. easy to understand - The variable name should be easy to understand. For example, the variable name "w" does not offer enough clarity. Different people may interpret it differently, like "weight", "width", "week", etc.

c. easy to debug

Following are some of the rules for creating a variable

1. A variable name must start with a letter or underscore character. It can not start with a number.

```
1weight = 56
```

```
SyntaxError: invalid syntax
```

In the above example, the variable name starts with a number. This does not follow the rule for naming the variable in Python, and therefore, Python raises a SyntaxError.

2. A variable name can only contain alphanumeric characters and underscores (A-Z, a-z, 0-9, and _).

```
revenue-expenses = 100
```

```
SyntaxError: can't assign to operator
```

In this example, the variable name contains a non-alpha-numeric character, i.e., "-". As per the rule for naming the variable in Python, the variable name can have only alphanumeric characters, and therefore, Python raises a SyntaxError.

3. Variable names are case-sensitive (height, Height, and HEIGHT are three different variables)

```
height = 158
print(HEIGHT)
```

```
NameError: name 'HEIGHT' is not defined
```

In the above example, variable height is created and in the print statement, HEIGHT is used. As in Python variables are case sensitive, Python raises a NameError. While naming the variable one should keep uniformity regarding the case (uppercase or lowercase) of the variable.

4. The reserved words (keywords) cannot be used as a variable name.

```
True = 21
```

```
SyntaxError: can't assign to keyword
```

In the above example, the name "True" is used to create a variable. As "True" is a keyword in Python and keywords cannot be used as variable names, Python raises a SyntaxError.

< Previous Next >