1.0-basic

December 18, 2024

0.0.1 Syntax and Semantics in Python

Video Outline: - Single line Comments and multiline comments - Definition of Syntax and Semantics - Basic Syntax Rules in Python - Understanding Semantics in Python - Common Syntax Errors and How to Avoid Them - Practical Code Examples

Syntax refers to the set of rules that defines the combinations of symbols that are considered to be correctly structured programs in a language. In simpler terms, syntax is about the correct arrangement of words and symbols in a code.

Semantics refers to the meaning or the interpretation of the symbols, characters, and commands in a language. It is about what the code is supposed to do when it runs.

[1]:

```python
## Basic Syntax Rules In Python
## Case sensitivity- Python is case sensitive

name="Mukesh"
Name="Kumar"

print(name)
print(Name)
```

```
Mukesh
Kumar
```

0.0.2 Indentation

Indentation in Python is used to define the structure and hierarchy of the code. Unlike many other programming languages that use braces {} to delimit blocks of code, Python uses indentation to determine the grouping of statements. This means that all the statements within a block must be indented at the same level.

[2]:

```python
## Indentation
## Python uses indentation to define blocks of code. Consistent use of spaces␣
 ↪(commonly 4) or a tab is required.

age=32
if age>30:

    print(age)
print(age)
```

```
32
32
```

[3]:

```
## This is a single line comment
print("Hello World")
```

```
Hello World
```

[5]:

```
## Line Continuation
##Use a backslash (\) to continue a statement to the next line

total=1+2+3+4+5+6+7+\
4+5+6

print(total)
```

```
43
```

[6]:

```
## Multiple Statements on a single line
x=5;y=10;z=x+y
print(z)
```

```
15
```

[2]:

```
##Understand  Semnatics In Python
# variable assignment
age=32 ##age is an integer
name="Mukesh" ##name is a string
```

[10]:

```
type(age)
```

[10]:

```
int
```

[9]:

```
type(name)
```

[9]:

```
str
```

[3]:

```
## Type Inference
variable=10
print(type(variable))
variable="Mukesh"
print(type(variable))
```

```
<class 'int'>
<class 'str'>
```

[14]:
```
age=32
if age>30:
    print(age)
```

```
32
```

[15]:
```
## Name Error
a=b
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 a=b

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

[17]:
```
## Code exmaples of indentation
if True:
    print("Correct Indentation")
    if False:
        print("This ont print")
    print("This will print")
print("Outside the if block")
```

```
Correct Indentation
This will print
Outside the if block
```

0.0.3 Conclusion:

Understanding the syntax and semantics of Python is crucial for writing correct and meaningful programs. Syntax ensures the code is properly structured, while semantics ensures the code behaves as expected. Mastering these concepts will help in writing efficient and error-free Python code.