Python Identifiers and Reserved Words

Identifiers

In Python, a **name used in a program** is called an *identifier*. It can be the name of a:

- Class
- Function
- Module
- Variable

Example:

```
a = 10
```

Rules for Defining Identifiers in Python

1. **Allowed Characters**
   Identifiers can only contain:
   - Alphabet symbols (a–z, A–Z)
   - Digits (0–9)
   - Underscore symbol `_`
2. ❌ Using any other symbol like `$` will result in a syntax error.

```
cash = 10   # Valid
ca$h = 20   # Invalid
```

3. **Should Not Start with a Digit**

```
123total = 50   # Invalid
total123 = 50   # Valid
```

4. **Case Sensitivity**
   Identifiers in Python are case-sensitive.

```
total = 10
TOTAL = 999

print(total)  # Output: 10
print(TOTAL)  # Output: 999
```

5. **Cannot Use Reserved Words**
   Reserved words in Python cannot be used as identifiers.

```
def = 10   # Invalid
```

6. **No Length Limit**
   There is no maximum length for an identifier, but it's best to avoid excessively long names.
7. **Dollar Symbol Not Allowed**
   `$` is not permitted in Python identifiers.
8. **Identifiers Starting with Underscore**
   - `_identifier`: Indicates *private*.
   - `__identifier`: Indicates *strongly private*.
   - `__identifier__`: Indicates a *language-defined special name* (also known as *magic methods*).
   - Example:

```
__add__  # Magic method used for operator overloading
```

Valid and Invalid Identifiers

Identifier	Validity
`123total`	❌ Invalid
`total123`	✅ Valid
`java2share`	✅ Valid
`ca$h`	❌ Invalid

Identifier	Validity
`_abc_abc_`	✅ Valid
`def`	❌ Invalid (reserved word)
`if`	❌ Invalid (reserved word)

Reserved Words

In Python, certain words are **reserved** to represent specific meanings or functionalities. These are known as **reserved words** or **keywords**.

There are 33 reserved words in Python (as of version 3.x):

```
True, False, None,
and, or, not, is,
if, elif, else,
while, for, break, continue, return, in, yield,
try, except, finally, raise, assert,
import, from, as, class, def, pass, global, nonlocal, lambda, del, with
```

Notes:

1. All reserved words contain **only alphabet characters**.
2. All are in **lowercase**, *except*:
   - `True`
   - `False`
   - `None`

```
a = true   # Invalid
a = True   # Valid
```

Checking Reserved Words in Python

You can view the list of reserved words using the `keyword` module:

```
import keyword
print(keyword.kwlist)
```

Output:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue',
 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global',
 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass',
 'raise', 'return', 'try', 'while', 'with', 'yield']
```