🔑 Keywords in Python

In Python, **keywords** are reserved words that have special meaning. These words are part of the Python syntax and **cannot** be used as variable names, function names, or identifiers.

Python has a fixed set of keywords that define the language's structure. These are always written in lowercase (except `True`, `False`, and `None`).

You can get the full list of keywords using the `keyword` module:

```
import keyword

# List all keywords in Python
print(keyword.kwlist)
print(f"Total number of keywords in Python: {len(keyword.kwlist)}")
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except',
Total number of keywords in Python: 35
```

❌ Don't Use Keywords as Variable Names

```
# Invalid example
if = 5       # ❌ This will raise a SyntaxError
class = "A"  # ❌ 'class' is a reserved keyword
```

✅ Valid Example

```
value = 5
my_class = "A"
```

✅ Tips

- Use `my_var`, `data_1`, `total_sum` instead of names like `if`, `else`, `for`, `try`, etc.
- Use `keyword.iskeyword("xyz")` to check if a word is a keyword or not.

```
# Check if a word is a keyword
import keyword

print(keyword.iskeyword("if"))     # True
print(keyword.iskeyword("hello"))  # False
```

```
True
False
```

```
Start coding or generate with AI.
```

🧠 Practice Exercise: Try it Yourself!

1. Try using any of these keywords in a small program: `if`, `else`, `for`, `while`.
2. Attempt to use a keyword as a variable name — observe the error.
3. Use the `keyword` module in Python to print all keywords.
4. Try creating a function named `def`, `class`, or `return` and understand what error you get.
5. Explore how `True`, `False`, and `None` behave in expressions.