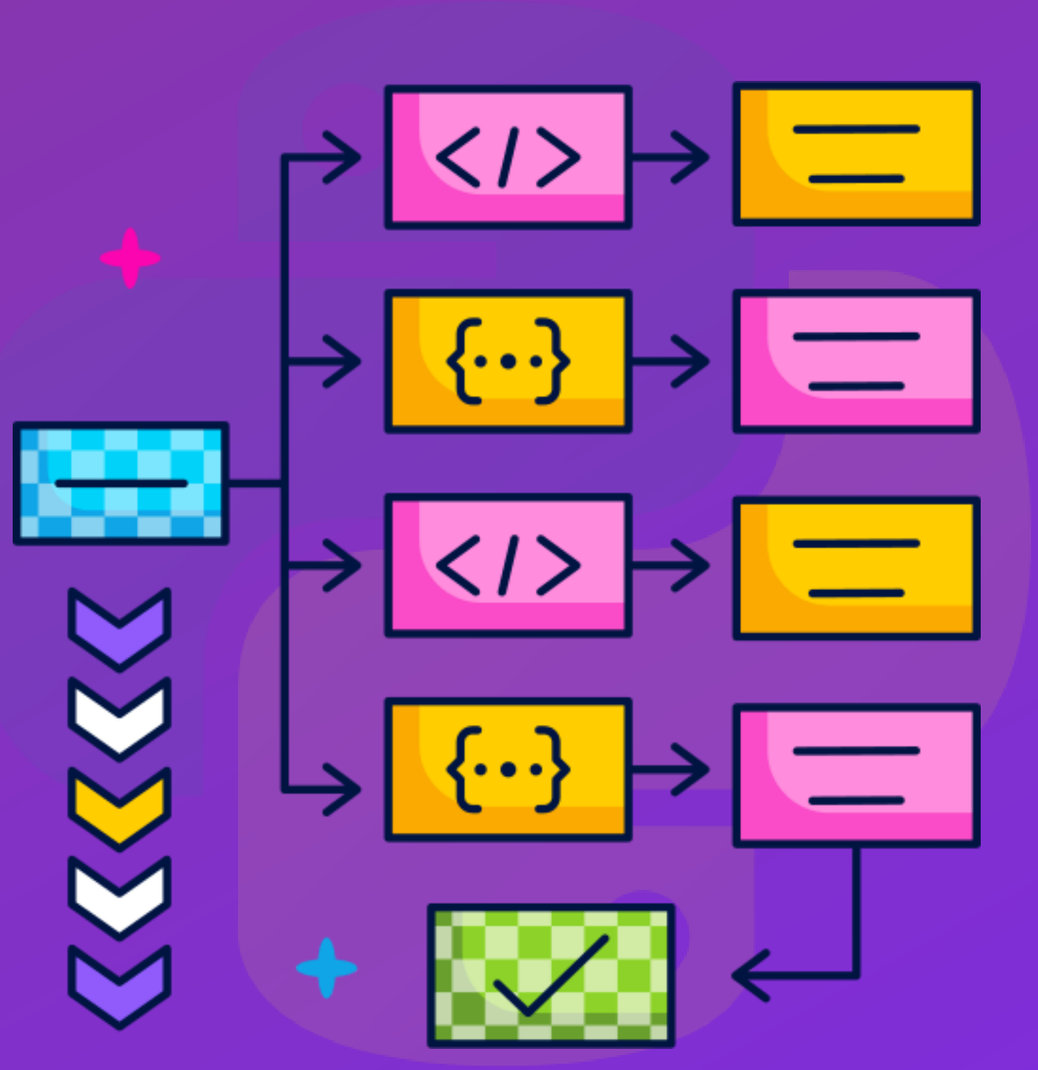
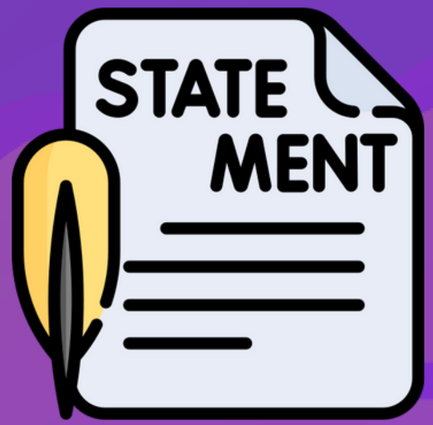
PROGRAM FLOW CONTROL



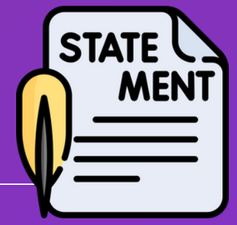
PROGRAM FLOW CONTROL

```
if  some_condition_is_true:
    # 1. execute_this_code
elif some_other_condition_is_true:
    # 2. execute_this_code
else:
    # 3. execute_this_code
```

STATEMENTS VS. EXPRESSIONS

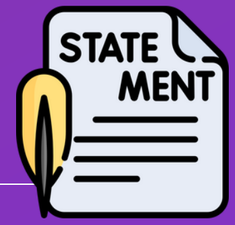


STATEMENTS VS. EXPRESSIONS



- **Statements** perform actions and control the flow or structure of the program. They don’t directly produce a value. Example: *x = 5*

STATEMENTS VS. EXPRESSIONS



- **Statements** perform actions and control the flow or structure of the program. They don’t directly produce a value. Example: *x = 5*

- **Expressions** always evaluate to a value. They can appear inside statements. Example: *len('Hello')*

BOOLEAN VARIABLES AND EXPRESSIONS

- A boolean variable is an object of the bool class which is an int subclass.

Boolean constants:

1. True
2. False

LOGICAL (BOOLEAN) OPERATORS: AND, OR, NOT

LOGICAL (BOOLEAN) OPERATORS

- A boolean expression is a logical statement that is either True or False.

LOGICAL (BOOLEAN) OPERATORS

- A boolean expression is a logical statement that is either True or False.

Logical (Boolean) operators:

1. and
2. or
3. not

THE LOGICAL “and” OPERATOR

- The “and” operator returns True only if both conditions are True.
- If any condition is False, the entire expression is False.

THE LOGICAL “or” OPERATOR

- With the “or” operator, only one of the conditions needs to be True for the whole expression to evaluate as True.

- If all conditions are False, the result is False.

THE BREAK STATEMENT

THE BREAK STATEMENT

- The break statement breaks out of the innermost enclosing for or while loop.

THE BREAK STATEMENT

- The break statement breaks out of the innermost enclosing for or while loop.

- If the break statement is inside a nested loop, break will terminate only the innermost loop.