# Python Basics Tutoring Material  
  
## Introduction to Python  
  
- \*\*What is Python?\*\*  
 - Python is a high-level, interpreted programming language known for its readability and simplicity.  
 - Widely used in web development, data analysis, artificial intelligence, scientific computing, and more.  
  
## Setting Up Python  
  
- \*\*Installation\*\*  
 - Download Python from the official website: [python.org](https://www.python.org/)  
 - Follow the installation instructions for your operating system.  
- \*\*Running Python\*\*  
 - Use the Python interactive shell by typing `python` or `python3` in your terminal.  
 - Run Python scripts by saving them with a `.py` extension and executing `python script.py`.  
  
## Basic Syntax  
  
- \*\*Variables and Data Types\*\*  
 - Variables are used to store data. Example: `x = 5`  
 - Common data types: `int`, `float`, `str`, `bool`  
- \*\*Operators\*\*  
 - Arithmetic: `+`, `-`, `\*`, `/`, `//`, `%`, `\*\*`  
 - Comparison: `==`, `!=`, `>`, `<`, `>=`, `<=`  
 - Logical: `and`, `or`, `not`  
  
## Control Structures  
  
- \*\*Conditional Statements\*\*  
  
<br>

# Python Control Structures  
  
## Introduction  
  
Control structures allow you to dictate the flow of your program. They enable decision-making, looping, and branching, which are essential for creating dynamic and responsive programs.  
  
## Conditional Statements  
  
### `if` Statement  
  
- \*\*Syntax\*\*:  
 ```python  
 if condition:  
 # code to execute if condition is true  
 ```  
  
- \*\*Example\*\*:  
 ```python  
 age = 18  
 if age >= 18:  
 print("You are eligible to vote.")  
 ```  
  
### `if-else` Statement  
  
- \*\*Syntax\*\*:  
 ```python  
 if condition:  
 # code if condition is true  
 else:  
 # code if condition is false  
 ```  
  
- \*\*Example\*\*:  
 ```python  
 age = 16  
 if age >= 18:  
 print("You are eligible to vote.")  
 else:  
 print("You are not eligible to vote.")  
 ```  
  
### `if-elif-else` Statement  
  
- \*\*Syntax\*\*:  
 ```python  
 if condition1:  
 # code if condition1 is true  
 elif condition2:  
 # code if condition2 is true  
 else:  
 # code if none of the above conditions are true  
 ```  
  
- \*\*Example\*\*:  
 ```python  
 score = 85  
 if score >= 90:  
 print("Grade: A")  
 elif score >= 80:  
 print("Grade: B")  
 elif score >= 70:  
 print("Grade: C")  
 else:  
 print("Grade: D")  
 ```  
  
## Loops  
  
### `for` Loop  
  
- \*\*Used for iterating over a sequence (like a list, tuple, or string).\*\*  
  
- \*\*Syntax\*\*:  
 ```python  
 for variable in sequence:  
 # code to execute for each item in sequence  
 ```  
  
- \*\*Example\*\*:  
 ```python  
 fruits = ["apple", "banana", "cherry"]  
 for fruit in fruits:  
 print(fruit)  
 ```  
  
### `while` Loop  
  
- \*\*Repeats as long as a condition is true.\*\*  
  
- \*\*Syntax\*\*:  
 ```python  
 while condition:  
 # code to execute as long as condition is true  
 ```  
  
- \*\*Example\*\*:  
 ```python  
 count = 0  
 while count < 5:  
 print(count)  
 count += 1  
 ```  
  
## Loop Control Statements  
  
### `break`  
  
- \*\*Terminates the loop prematurely.\*\*  
  
- \*\*Example\*\*:  
 ```python  
 for number in range(10):  
 if number == 5:  
 break  
 print(number)  
 ```  
  
### `continue`  
  
- \*\*Skips the current iteration and moves to the next one.\*\*  
  
- \*\*Example\*\*:  
 ```python  
 for number in range(10):  
 if number % 2 == 0:  
 continue  
 print(number)  
 ```  
  
### `pass`  
  
- \*\*A null operation; it does nothing. Useful as a placeholder.\*\*  
  
- \*\*Example\*\*:  
 ```python  
 for number in range(5):  
 if number == 3:  
 pass # Placeholder for future code  
 print(number)  
 ```  
  
## Practical Exercise  
  
1. \*\*Conditional Exercise\*\*: Write a program that checks if a number is positive, negative, or zero and prints an appropriate message.  
  
2. \*\*Loop Exercise\*\*: Create a loop that prints all the even numbers from 1 to 20.  
  
3. \*\*Loop Control Exercise\*\*: Modify the loop to skip the number 10 using `continue` and stop the loop entirely if the number reaches 16 using `break`.  
  
---  
  
This material provides a foundational understanding of control structures in Python, with examples and exercises to reinforce learning.