Python Fundamentals: Key Concepts and Code Examples

Strings

Strings are sequences of characters in Python. They can hold letters, words, or even entire sentences. Strings are created by enclosing characters in single quotes ' ', double quotes " ", or triple quotes ''' ''' for multi-line strings.

Example:

```
greeting = "Hello"
name = "Alice"
full_greeting = greeting + " " + name
print(full_greeting)                # Output: Hello Alice
print(len(full_greeting))           # Output: 11
print(greeting[0])                  # Output: H
print(greeting[1:4])                # Output: ell
```

Program Design and Development

Program design is the process of planning a program before coding. It often involves breaking the problem into smaller tasks, designing functions for each task, and testing iteratively.

Example:

```
def calculate_area(length, width):
    return length * width

length = 5
width = 3
area = calculate_area(length, width)
```

```
print("Area of rectangle:", area)      # Output: Area of rectangle: 15
```

Iteration

Iteration allows you to repeat a block of code multiple times. Python provides loops like `for` and `while` for iteration.

Example using `for` loop:

```
numbers = [1, 2, 3, 4, 5]
for number in numbers:
    print(number)
```

Example using `while` loop:

```
count = 5
while count > 0:
    print(count)
    count -= 1
```

Conditionals

Conditionals allow you to execute code based on a condition. The `if`, `elif`, and `else` statements are used to define conditions.

Example:

```
age = 20
if age >= 18:
    print("You are an adult.")
elif age >= 13:
```

```
    print("You are a teenager.")
else:
    print("You are a child.")
```

Lists

Lists are used to store multiple items in a single variable. Lists are mutable, meaning you can change their content after creation.

Example:

```
fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
print(fruits)                    # Output: ['apple', 'banana', 'cherry', 'orange']
print(fruits[1])                 # Output: banana
fruits.remove("banana")
print(fruits)                    # Output: ['apple', 'cherry', 'orange']
```

Nested Conditionals

Nested conditionals involve using `if` statements inside other `if` statements, allowing for more complex decision-making.

Example:

```
x = 10
y = 20
if x > 5:
    if y > 15:
        print("x is greater than 5 and y is greater than 15.")
```

```
    else:
        print("x is greater than 5 but y is not greater than 15.")
else:
    print("x is not greater than 5.")
```