Tutorial 1: Introduction to Python - Let's Start Coding! 🚀

Table of Contents

1. What is Python? 🐍

What Makes Python Special?

Python is a programming language created to be easy to read and understand. Imagine giving instructions to a friend - Python lets you write code that almost reads like regular English! It's like having a conversation with your computer.

Why Learn Python?

- **Beginner-Friendly**: Python uses simple, clear commands
- **Powerful**: Despite being easy to learn, it can create amazing things
- **Widely Used**: Companies like Google, Netflix, and NASA use Python
- **Versatile**: You can make games, websites, apps, and much more!

What Can You Build?

- Games like simple quiz games or adventure games
- Programs that can solve math problems
- Tools to organize your files
- Websites and web applications
- Data analysis and visualization

Setting Up Python 🛠️

Before we start coding, we need to install Python on your computer:

1. **Download Python**:
    - Go to python.org
    - Click "Downloads"
    - Download the latest version for your operating system
2. **Install Python**:

- Run the installer you downloaded
- Important: Check ✅ "Add Python to PATH"
- Click "Install Now"

3. **Choose an Editor**:

- IDLE (comes with Python) - great for beginners
- Thonny [https://thonny.org] - another beginner-friendly IDE
- PyCharm Educational Edition - more features but still beginner-friendly
- Visual Studio Code - very popular but might be complex at first

2. Your First Python Program 👋

Understanding Code Basics

Just like learning a new language, we'll start with something simple. In Python, we use the `print()` command to show text on the screen.

```
# This is a comment - Python ignores anything after #
# Comments help us explain what our code does

print("Hello, World!")  # This shows text on the screen
```

How It Works:

1. `print()` is a command (we call it a function)
2. The text inside quotes `"Hello, World!"` is what we want to show
3. When you run this program, it displays: `Hello, World!`

Try These Examples:

```
# Simple messages
print("Welcome to Python!")
print("I'm learning to code!")

# Multiple lines
print("First line")
print("Second line")
print("Third line")

# Printing numbers (no quotes needed for numbers)
print(42)
print(3.14)
```

Common Mistakes to Avoid:

```
# Wrong: Missing quotes
print(Hello World)    # This will cause an error

# Wrong: Missing parentheses
print "Hello"         # This will cause an error

# Correct:
print("Hello World")
```

3. Variables and Data Types 📦

Understanding Variables

Think of variables as labeled containers that store information:

- Like a box labeled "age" that holds a number
- Or a box labeled "name" that holds text
- You can change what's in the box anytime

Types of Data

1. Numbers

Python has different types of numbers:

```
# Integers (whole numbers)
age = 15             # An integer (int)
students = 30
negative = -5

# Decimal numbers (floating-point)
height = 1.75        # A float
temperature = 98.6
pi = 3.14159

# You can do math with both types
age = 15
next_year = age + 1
print(next_year)     # Shows: 16
```

2. Strings (Text)

Any text between quotes is a string:

```
# Different ways to create strings
name = "Alex"        # Double quotes
color = 'blue'       # Single quotes
message = """This is
```

```
a multi-line
string"""           # Triple quotes for multiple lines

# Combining strings
first = "Hello"
last = "World"
full = first + " " + last    # Joining strings
print(full)         # Shows: Hello World
```

3. Booleans (True/False)

Booleans are like switches - they're either True or False:

```
is_raining = True
has_homework = False
game_over = False

# You can change boolean values
game_over = True    # Game is now over
```

4. Working with Numbers 🔢

Basic Operations

```
# Addition (+)
sum = 10 + 5
print(sum)      # Shows: 15

# Subtraction (-)
difference = 10 - 5
print(difference)   # Shows: 5

# Multiplication (*)
product = 10 * 5
print(product)    # Shows: 50

# Division (/)
quotient = 10 / 5
print(quotient)   # Shows: 2.0  (always gives a decimal number)
```

Special Operations

```
# Integer Division (//)
result = 17 // 5     # Divides and rounds down
print(result)        # Shows: 3

# Modulus (%)
```

```
remainder = 17 % 5   # Shows remainder after division
print(remainder)     # Shows: 2

# Exponents (**)
square = 5 ** 2     # 5 squared
cube = 2 ** 3       # 2 cubed
print(square)       # Shows: 25
print(cube)         # Shows: 8
```

Real World Examples

```
# Calculate total cost
pencils = 5
pencil_price = 0.75
total = pencils * pencil_price
print(f"Total cost: ${total}")    # Shows: Total cost: $3.75

# Convert minutes to hours and minutes
total_minutes = 145
hours = total_minutes // 60
minutes = total_minutes % 60
print(f"{total_minutes} minutes is {hours}h {minutes}m")
```

5. Working with Strings 📝

Creating Strings

```
# Using single or double quotes
name1 = "Alex"        # Double quotes
name2 = 'Alex'        # Single quotes

# Quotes inside strings
message1 = "Don't forget to smile"    # Using double quotes outside
message2 = 'He said "Hello!" to me'   # Using single quotes outside

# Multi-line strings
long_text = """This is a long message
that can span across
multiple lines!"""
```

String Operations

```
# Combining strings (concatenation)
first_name = "Alex"
last_name = "Smith"
full_name = first_name + " " + last_name
print(full_name)    # Shows: Alex Smith
```

```
# Repeating strings
print("Hip " * 2 + "Hooray!")    # Shows: Hip Hip Hooray!
print("-" * 20)                  # Shows: --------------------

# F-strings (formatted strings)
name = "Alex"
age = 15
print(f"My name is {name} and I am {age} years old")
```

String Methods

```
message = "Hello, Python!"

# Converting case
print(message.upper())      # Shows: HELLO, PYTHON!
print(message.lower())      # Shows: hello, python!

# Finding and counting
print(message.count('o'))   # Shows: 2
print(message.find('Python'))  # Shows: 7

# Removing whitespace
text = "   Hello   "
print(text.strip())        # Shows: Hello
```

6. Getting Input from Users 🎤

Basic Input

```
# Getting text input
name = input("What's your name? ")
print(f"Hello, {name}!")

# Getting numbers (remember to convert!)
age_string = input("How old are you? ")
age = int(age_string)
print(f"Next year you'll be {age + 1}")
```

Converting Input Types

```
# Converting to integer
height_cm = int(input("Height in cm: "))

# Converting to float
weight_kg = float(input("Weight in kg: "))
```

```
# Combining input and conversion
temperature = float(input("Temperature: "))
```

7. Organizing Your Python Programs 📂

File and Folder Organization (Windows)

Create a clear folder structure:

```
Python_Projects\
├── Tutorial_1\
│   ├── hello_world.py
│   └── exercises\
├── Tutorial_2\
└── Practice\
```

Naming Your Files

Good names:

```
hello_world.py
temperature_converter.py
calculator.py
```

Bad names:

```
program.py           # Too vague
My Program.py       # Has spaces
test.py             # Not descriptive
```

8. Type Conversion 🔄

Converting Strings to Numbers

```
# String to integer
text_number = "123"
number = int(text_number)    # number is now 123

# String to float
text_decimal = "12.34"
decimal = float(text_decimal)  # decimal is now 12.34
```

Converting Numbers to Strings

```python
# Number to string
age = 15
message = "I am " + str(age) + " years old"

# Decimal to string
price = 19.99
print("The price is $" + str(price))
```

9. Practice Exercises 🎯

Basic Print and Variables

1. Personal Introduction

Create a program that introduces yourself using variables and print statements.

```python
# Sample output:
# Hi! My name is Alex Smith
# I am 15 years old
# I love programming with Python! 🐍
```

2. Variable Swap

Create a program that swaps the values of two variables.

```python
# Sample code start:
a = 5
b = 10
# Your code here to swap values
# After swapping, a should be 10 and b should be 5
```

3. ASCII Art

Create a simple picture using text characters in multiple print statements.

```python
# Sample output (house):
#    /\
#   /  \
#  /____\
#  |    |
#  |  []|
#  |    |
#  ------
```

Number Operations

4. Rectangle Calculator

Write a program that:

- Asks for length and width of a rectangle
- Calculates both area and perimeter
- Prints results nicely formatted

```
# Sample run:
Enter length: 5
Enter width: 3
Area: 15 square units
Perimeter: 16 units
```

5. Temperature Converter

Create a program that converts temperature between Celsius and Fahrenheit. Formula: (C × 9/5) + 32 = F

```
# Sample run:
Enter temperature in Celsius: 30
30°C is equal to 86°F
```

6. Circle Calculator

Write a program that calculates the area and circumference of a circle. Use π (pi) = 3.14159

```
# Sample run:
Enter circle radius: 5
Area: 78.54 square units
Circumference: 31.42 units
```

7. Average Calculator

Create a program that:

- Takes three test scores
- Calculates their average
- Prints the result

```
# Sample run:
Enter first score: 85
Enter second score: 92
Enter third score: 88
Average score: 88.33
```

String Operations

8. Name Formatter

Create a program that:

- Takes first and last name
- Prints them in different formats

```
# Sample run:
Enter first name: Alex
Enter last name: Smith
Full name: Alex Smith
Reverse order: Smith, Alex
Initials: A.S.
```

9. String Repeater

Write a program that:

- Takes a word and a number
- Repeats the word that many times

```
# Sample run:
Enter a word: Hello
Enter number of repeats: 3
HelloHelloHello
```

10. Receipt Generator

Create a simple receipt that:

- Takes store name
- Takes three items and prices
- Displays them neatly formatted

```
# Sample run:
Enter store name: My Shop
Enter item 1: Apple
Enter price 1: 0.50
Enter item 2: Bread
Enter price 2: 1.25
Enter item 3: Milk
Enter price 3: 2.75

====================
      MY SHOP
====================
Apple:      $0.50
```

```
Bread:      $1.25
Milk:       $2.75
--------------------
Total:      $4.50
====================
```

Additional Math Exercises

11. Paint Calculator

Calculate how much paint is needed to paint a room:

- Get room dimensions (length, width, height)
- Calculate wall area (excluding ceiling)
- One liter of paint covers 10 square meters

```
# Sample run:
Enter room length (meters): 4
Enter room width (meters): 3
Enter room height (meters): 2.5

Wall area: 35.0 square meters
Paint needed: 3.5 liters
```

12. Money Converter

Create a program that converts a large amount of cents into dollars and cents:

```
# Sample run:
Enter amount in cents: 1234
That equals: $12.34
```

13. Distance and Time

Create a program that calculates average speed:

- Ask for distance in kilometers
- Ask for time in hours and minutes
- Calculate speed in km/h

```
# Sample run:
Enter distance (km): 150
Enter hours: 2
Enter minutes: 30
```

```
Time taken: 2.5 hours
Average speed: 60.0 km/h
```

14. Exercise Points Calculator

Create a program that calculates total exercise points:

- Running: 5 points per minute
- Swimming: 7 points per minute
- Cycling: 4 points per minute

```
# Sample run:
Minutes spent running: 20
Minutes spent swimming: 15
Minutes spent cycling: 30

Points earned:
Running: 100 points
Swimming: 105 points
Cycling: 120 points
Total points: 325
```

Type Conversion Exercises

15. Number Formatter

Create a program that:

- Takes a number
- Shows it in different formats

```
# Sample run:
Enter a number: 123.456
As integer: 123
With 2 decimal places: 123.46
As scientific notation: 1.23e+02
```

16. Shopping Calculator

Create a program that:

- Asks for prices of 5 items
- Calculates total
- Calculates average price per item
- Shows both rounded and exact values

```
# Sample run:
Enter price for item 1: 2.99
Enter price for item 2: 1.50
Enter price for item 3: 3.25
Enter price for item 4: 4.75
Enter price for item 5: 2.50

Receipt Summary:
Total (rounded): $15
Total (exact): $14.99
Average per item (rounded): $3
Average per item (exact): $2.998
```

Tips for Solving Exercises:

1. Read the problem carefully
2. Write down what inputs you need
3. Plan your calculations
4. Format your output nicely
5. Test with different values
6. Check your results by hand

Common Mistakes to Watch For:

1. Forgetting to convert input to numbers
2. Missing spaces in output formatting
3. Not using clear variable names
4. Forgetting to use f-strings for formatting
5. Not testing with different inputs

Remember:

- Start with the easier exercises
- Test your code frequently
- Keep your code neat and organized
- Add comments to explain your code
- Have fun while coding!

Tips for Success 💡

1. Always save your code files with `.py` extension
2. Use meaningful variable names
3. Add comments to explain your code
4. Test your program with different inputs
5. Keep your code organized and neat
6. Ask for help when stuck!

Common Mistakes to Avoid ⚠️

1. Forgetting quotes around strings
2. Not converting input to the right type
3. Using spaces in file names
4. Forgetting to save files before running
5. Missing parentheses in print()

Remember: Programming is like learning a new language - practice makes perfect! Don't be afraid to experiment and make mistakes. That's how we learn!

Need help? Ask your teacher or try explaining your code to a friend - sometimes just talking about it helps you find the solution!

Happy coding! 🚀