

- **High-Level vs. Low-Level Languages:** High-level languages are easier to read and write, portable across systems, and abstract hardware details, while low-level languages are closer to machine code and specific to hardware.

- **Translators:** Interpreters execute code line by line, while compilers translate the entire code into machine language before execution.

- **Variables:** Variables store data values. Example: `x = 10`.

- **Indentation:** Indentation defines code blocks in Python, such as loops, conditionals, or functions.

- **Function Definition:** Functions are defined using `def`. Example:

```
python
Copy code
def add(a, b):
    return a + b
```

- **Loops:** A for-loop iterates over sequences. Example:

```
python
Copy code
for char in "Python":
    print(char)
```

- **Error Handling:** Referencing an undefined variable raises a `NameError`.

- **Modulo Operator:** `%` returns the remainder of a division. Example: `10 % 2 == 0` indicates an even number.

- **Pixel Manipulation:** Change a pixel's color using functions like `getPixel()` and `setColor()`.

- **Green Screen Effect:** Replace green pixels with background pixels by comparing RGB values.