

Python + Pygame Cheat Sheet

Python Basics for Game Development

Python is a programming language that's great for beginners. It's used in web development, data science, and you guessed it game development!

What is a Variable?

A variable stores information like a number or a word so you can use it later.

```
x = 10 # This stores the number 10 in a variable called x
name = "snake" # This stores the word 'snake' in a variable
```

What is a List?

A list lets you store multiple things in one variable. Think of it like a container of items.

```
items = [1, 2, 3]
items.append(4) # Adds 4 to the list
```

If Statements (Making Decisions)

Use 'if' to tell the computer to do something only when a condition is true.

```
if x > 5:
    print("Big number!")
```

Loops (Doing Things Repeatedly)

Loops repeat actions. A 'for' loop counts. A 'while' loop runs until something changes.

```
for i in range(5):
    print(i)
```

```
while running:
    do_something()
```

Functions (Reusable Code)

A function is like a recipe. You define it once, then use it whenever you want.

```
def greet(name):
    print("Hi", name)
```

```
greet("Alex") # This prints: Hi Alex
```

Random Numbers

You can make your game unpredictable using random numbers.

```
import random
random.randint(1, 10) # Gives a random number from 1 to 10
```

Python + Pygame Cheat Sheet

Pygame Basics

Pygame is a Python library that helps you make 2D games. It handles drawing, timing, and keyboard input.

Starting Pygame

Always start your game by initializing Pygame and setting up the screen.

```
import pygame
pygame.init()
screen = pygame.display.set_mode((WIDTH, HEIGHT))
clock = pygame.time.Clock()
```

Drawing on the Screen

You can draw shapes like rectangles (used to make your snake!)

```
pygame.draw.rect(screen, (0, 255, 0), (x, y, width, height)) # Green square
```

Handling Input (Keyboard)

Let the player control the game with arrow keys or other inputs.

```
keys = pygame.key.get_pressed()
if keys[pygame.K_LEFT]:
    direction = (-1, 0) # Move left
```

Game Loop

This is the heart of your game. It runs repeatedly to update and redraw everything.

```
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    pygame.display.flip()
    clock.tick(10)
```

Collisions

To check if things touch each other (like your snake and food), use rectangles.

```
rect1 = pygame.Rect(x1, y1, w1, h1)
rect2 = pygame.Rect(x2, y2, w2, h2)
if rect1.colliderect(rect2):
    print("They touched!")
```

Python + Pygame Cheat Sheet

Python Basics for Game Development (Extended)

Python is simple to read and powerful for beginners. These concepts help you build game logic efficiently.

List Slicing

You can grab parts of a list using slices. Think of slicing like cutting a piece of bread.

```
items = [10, 20, 30, 40, 50]
print(items[1:4]) # [20, 30, 40]
```

Negative Indexes

Negative indexes let you access the list from the end. -1 is the last item.

```
print(items[-1]) # 50 (last item)
print(items[-2]) # 40
```

Multiple Variable Assignment

Python lets you assign values to multiple variables at once.

```
x, y = 5, 10
print(x) # 5
print(y) # 10
```

Python + Pygame Cheat Sheet

Pygame Vocabulary

Here are some key Pygame terms you'll see in your projects:

screen:

The window where your game is drawn.

clock:

Controls how fast your game runs (frames per second).

event:

Something that happens, like a key being pressed.

rect:

A rectangle used for drawing and collision detection.

flip():

Updates the game window with any new drawings.

key.get_pressed():

Checks if any keys are currently being held down.

QUIT:

An event that happens when you click the close button.