GUIDED NOTES

Welcome Back: The Python You Know

Name: _____ Date: _____

Fill in the blanks as we go. Each day has its own word bank, and every answer is in it.

WHAT YOU ALREADY KNOW

`int(input())` turns a typed answer into a whole number, and an f-string puts a value inside text. With `age = int(input("Age? "))`, typing 15 makes `print(f"Next year you will be {age + 1}.")` print Next year you will be 16.

An `if` chain checks its tests from the top and runs only the first one that's `True`. With `score = 75`, a chain that prints Gold for `score >= 90`, Silver for `score >= 70` and Bronze for the rest prints Silver.

A loop can add up numbers. Start with `total = 0`. The loop `for lap in range(1, 4):` with `total += lap` inside it adds 1, 2 and 3, so `print(total)` prints 6.

Day 1 — Update Operators and Editor Tools

Today you will learn how an update operator changes a variable in one step, and how the editor helps you write and fix code.

Word bank auto-indent · comment out · line number · suggestion list · unary minus · update operator

Update operators

An update operator changes a variable in one step.

```
coins = 20
coins //= 3
coins *= 4
coins %= 5
coins **= 2
coins /= 2
print(coins)
```

Output

```
8.0
```

An _____ changes a variable using its old value. `coins //= 3` is short for `coins = coins // 3`.

`*=` multiplies, `/=` divides and `//=` divides and rounds down to a whole number. `%=` keeps the remainder, and `**=` raises to a power.

`/=` always gives a decimal, like 8.0.

The order of math

`**` goes before a minus sign in front.

```
print(2 ** 3 ** 2)
print(-2 ** 2)
```

Output

```
512
-4
```

`**` works right to left. In `2 ** 3 ** 2`, Python works out `3 ** 2` first.

A _____ is a minus sign in front of one value. `**` goes first, so `-2 ** 2` is -4.

Read the Error panel

An error names the line to fix.

```
1  score = 8
2  score *= 3
3  print(scor)
```

Error

Line 3: Python says: `NameError: name 'scor' is not defined`

Each line has a _____ at its left. The Error panel names the line where Python stopped.

A `NameError` often means a name is spelled wrong.

Editor tools

The editor helps you write and fix code.

```
1  laps = 0
2  # laps = 50
3  for run in range(4):
4      laps += 2
5  print(laps)
```

Output

```
8
```

To _____ a line, type # in front of it. Python skips the line until you delete the #.

With _____, pressing Enter after a colon starts the next line 4 spaces in.

The _____ shows words as you type. Pick one with Tab or Enter, or tap it.

REMEMBER

An update operator changes a variable using its old value, in one step. ** works right to left, and it goes before a minus sign in front. The Error panel names the line to fix, and a # turns a line off.

Day 2 — Fix and Finish with Editor Tools

Today you will learn how to use update operators and the editor's tools to fix and finish short programs.

Word bank auto-indent · comment out · Error panel · indent · remainder · test line

Whole groups and leftovers

`//` counts full groups, and `%=` keeps the rest.

```
1  bars = 11
2  bags = bars // 4
3  bars %= 4
4  print(f"{bags} bags, {bars} left")
```

Output

```
2 bags, 3 left
```

`bars // 4` counts the full bags of 4. It divides and rounds down to a whole number, so 11 bars make 2 bags.

The _____ is what's left over after a division. `bars %= 4` keeps it, so `bars` goes from 11 to 3.

Indents and auto-indent

A line after a colon starts 4 spaces in.

```
1  seconds = 40
2  while seconds > 1:
3  seconds //= 2
4  print(seconds)
```

Error

Line 3: Python says: `IndentationError: expected an indented block after 'while' statement on line 2`

An _____ is the spaces at the start of a line. The body of a loop or an `if` starts 4 spaces in.

A missing indent stops the program with an `IndentationError`.

With _____, the editor adds the 4 spaces when you press Enter after a colon.

Fixing and testing

The Error panel and comments help you fix code.

```
1  total = 12
2  # print("test:", total)
3  print(f"Total: {total}")
```

Output

```
Total: 12
```

The _____ names the line where Python stopped. Fix that line first.

A _____ prints or sets a value while you work.

To _____ a test line, type # in front of it. Delete the # to turn it back on.

REMEMBER

// counts full groups, and **%=** keeps the remainder. The Error panel names the line to fix, auto-indent adds the 4 spaces after a colon and a **#** turns a test line off.

Day 3 — Find the Bonus Bug

Today you will review update operators and the editor's tools, fix a bug that gives a wrong answer and take the lesson quiz.

Word bank Error panel · Exponent order · logic error · SyntaxError · update operator

Update operators

An update operator uses the old value.

```
1  laps = 10
2  laps //= 3
3  laps **= 2
4  print(laps)
```

Output

```
9
```

An _____ like `//=` or `**=` uses a variable's old value and stores a new one.

`laps //= 3` does the same job as `laps = laps // 3`.

Exponent order

`**` works right to left.

```
1  print(2 ** 2 ** 3)
```

Output

```
256
```

_____ is right to left, so `2 ** 2 ** 3` is `2 ** 8`, or 256.

`**` also goes before a minus sign in front.

Two kinds of bugs

Some bugs stop the program, and some don't.

```
1  tricks = 4
2  if tricks >= 3
3      print("Bonus round!")
```

Error

Line 2: Python says: `SyntaxError: expected ':'`

A _____ means Python can't read a line. The _____ names that line.

A _____ gives a wrong answer with no error message. Compare the output with what it should be.

REMEMBER

An update operator uses a variable's old value to store a new one, and ** works right to left. A SyntaxError stops the program at the line the Error panel names, and a logic error gives a wrong answer with no message.