

# 1.Answers – Variables and Basic Data Types

# Integer

num = 6 # **Correct**

num1 = "10" # **Incorrect** - This is a string, not an integer

2num = 10 # **Incorrect** - variable cannot start with number

num3 = 3.14 # **Incorrect** - This is a float, not an integer

num4 = -5 # **Correct**

num6 = True # **Incorrect** -This is a boolean, not an integer

# Float

float1 = "10.0" # **Incorrect** - This is a string, not a float

float2 = 8.0 # **Correct** float

float3 = 8 # **Incorrect** - This is an integer, not a float

float4 = 3 # **Incorrect** - This is an integer, not a float

float5 = -5.5 # **Correct** float

float6 = 0.0 # **Correct** float

float7 = False # **Incorrect** - This is a boolean, not a float

float8 = 2.0 # # **Incorrect** - variable indentation is **Incorrect** (4 spaces or tab)

pi = 3.14 # **Correct**

# String

str1 = "Hello, World!" # **Correct** string

str2 = 123 # **Incorrect** - This is an integer, not a string

str3 = 3.14 # **Incorrect** - This is a float, not a string

str4 = 'Python is fun.' # **Correct** string

str5 = """This is a

multi-line string.""" # **Correct** string

7str = "I am 7th!" # **Incorrect** - variable cannot start with number

Str8 = True # **Incorrect** - This is a boolean, not a string

# Boolean

Bool1 = "True" # **Incorrect** - # This is a string, not a boolean

Bool2 = True # **Correct** – typical way

Bool3 = 1 # **Incorrect** - This is an integer, not a boolean

Bool4 = False # **Correct** – typical way

Bool5 = 0.0 # **Incorrect** - This is a float, not a boolean

Bool6 = bool(1) # **Correct** returns True

Bool7 = bool(-1) # **Correct** returns True

Bool8 = bool(5) # **Correct** returns True

Bool9 = bool(0) # **Correct** returns False

Bool10 = 3 == 2 # **Correct** returns False

# 2.Group Work - Arithmetic Operators (3-4 people in a group)

7 minutes in a group #**Correct** or #**Incorrect** (if possible – why?)

4 minutes check against answers (one page without answers | 4 with answers to keep)

# Addition

add = 5 + 3

add1 = "5" + 3

add2 = 5 + "3"

add3 = "5" + "3"

add4 = 5.0 + 3

add5 = True + 3

# Subtraction

sub = 5 - 3

sub1 = "5" - 3

sub2 = 5 - "3"

sub3 = "5" - "3"

sub4 = 5.0 - 3

sub5 = True - 3

# Modulus

mod = 10 % 3

mod1 = "10" % 3

mod2 = 10 % "3"

mod3 = "10" % "3"

mod4 = 10.0 % 3

mod5 = True % 3

# Division

div = 10 / 3

div1 = "10" / 3

div2 = 10 / "3"

div3 = "10" / "3"

div4 = 10.0 / 3

div5 = True / 3

div6 = 10 / 3

# Multiplication

mul = 5 \* 3

mul1 = "5" \* 3

mul2 = 5 \* "3"

mul3 = "5" \* "3"

mul4 = 5.0 \* 3

mul5 = True \* 3

# Exponentiation

exp = 2 \*\* 3

exp1 = "2" \*\* 3

exp2 = 2 \*\* "3"

exp3 = "2" \*\* "3"

exp4 = 2.0 \*\* 3

exp5 = True \*\* 3

# Floor Division

floor\_div = 9 // 3

floor\_div1 = "10" // 3

floor\_div2 = 10 // "3"

floor\_div3 = "10" // "3"

floor\_div4 = 9.0 // 2

floor\_div5 = True // 3

# 2.Answers - Arithmetic Operators

# Addition

add = 5 + 3 # **Correct**

add1 = "5" + 3 # **Incorrect** - can't add string and integer

add2 = 5 + "3" # **Incorrect** - can't add integer and string

add3 = "5" + "3" # **Correct** - concatenates strings # "53"

3add = "5" + "3" # **Correct** - concatenates strings # "53"

add4 = 5.0 + 3 # **Correct** - adds float and integer

add5 = True + 3 # **Incorrect** - can't add boolean and integer

# Subtraction

sub = 5 - 3 # **Correct**

sub1 = "5" - 3 # **Incorrect** - can't subtract integer from string

sub2 = 5 - "3" # **Incorrect** - can't subtract string from integer

sub3 = "5" - "3" # **Incorrect** - can't subtract strings

sub4 = 5.0 - 3 # **Correct** - subtracts integer from float

sub5 = True - 3 # **Incorrect** - can't subtract integer from boolean

# Modulus

mod = 10 % 3 # **Correct** - returns the remainder of the division, 3 goes 3 times in 10 and 1 is left = result 1

mod1 = "10" % 3 # **Incorrect** - can't use modulus with string and integer

mod2 = 10 % "3" # **Incorrect** - can't use modulus with integer and string

mod3 = "10" % "3" # **Incorrect** - can't use modulus with strings

mod4 = 10.0 % 3 # **Correct** - returns the remainder of the division,3 goes 3 times in 10 and 1 is left = result 1

mod5 = True % 3 # **Incorrect** - can't use modulus with boolean and integer

# Division

div = 10 / 3 # **Correct** - performs floating point division - 3.3333

div1 = "10" / 3 # **Incorrect** - can't divide string by integer

div2 = 10 / "3" # **Incorrect** - can't divide integer by string

div3 = "10" / "3" # **Incorrect** - can't divide string by string

div4 = 10.0 / 3 # **Correct** - performs floating point division - 3.3333

div5 = True / 3 # **Incorrect** - can't divide boolean by integer

div6 = 10 / 3 # # **Incorrect** - variable indentation is **Incorrect** (4 spaces or tab)

# Multiplication

mul = 5 \* 3 # **Correct** - multiplies two numbers

mul1 = "5" \* 3 # **Correct** - repeats the string 3 times - "555"

mul2 = 5 \* "3" # **Correct** - repeats the string 5 times - "33333"

mul3 = "5" \* "3" # **Incorrect** - can't multiply strings

mul4 = 5.0 \* 3 # **Correct** - multiplies float and integer

mul5 = True \* 3 # **Incorrect** - can't multiply boolean and integer

# Exponentiation

exp = 2 \*\* 3 # **Correct** - raises the first number to the power of the second number

exp1 = "2" \*\* 3 # **Incorrect** - can't raise string to the power of integer

exp2 = 2 \*\* "3" # **Incorrect** - can't raise integer to the power of string

exp3 = "2" \*\* "3" # **Incorrect** - can't raise string to the power of string

exp4 = 2.0 \*\* 3 # **Correct** - raises float to the power of integer

exp5 = True \*\* 3 # **Incorrect** - can't raise boolean to the power of integer

# Floor Division

floor\_div = 9 // 3 # **Correct** - performs integer division, rounding down - 3

floor\_div1 = "10" // 3 # **Incorrect** - can't perform floor division with string and integer

floor\_div2 = 10 // "3" # **Incorrect** - can't perform floor division with integer and string

floor\_div3 = "10" // "3" # **Incorrect** - can't perform floor division with strings

floor\_div4 = 9.0 // 2 # **Correct** - performs floor division, rounding down - 3

floor\_div5 = True // 3 # **Incorrect** - can't perform floor division with boolean and integer

# 3.Group Work - Comparison Operators (3-4 people in a group)

7 minutes in a group #**Correct** or #**Incorrect** (if possible – why?)

4 minutes check against answers (one page without answers | 4 with answers to keep)

# Equality

eq = 5 == 3

eq1 = "5" == 3

eq2 = 5 == "3"

eq3 = "5" == "3"

eq4 = 5.0 == 3

eq5 = True == 3

# Not Equal

neq = 5 != 3

neq1 = "5" != 3

neq2 = 5 != "3"

neq3 = "5" != "3"

neq4 = 5.0 != 3

neq5 = True != 3

# Greater than

gt = 5 > 3

gt1 = "5" > 3

gt2 = 5 > "3"

gt3 = "5" > "3"

gt4 = 5.0 > 3

gt5 = True > 3

# other similar comparison operators:

# Less than <

# Greater than or equal >=

# Less than or equal <=

# 3.Answers - Comparison Operators

# Equality

eq = 5 == 3 # **Correct** - compares integers

eq1 = "5" == 3 # **Correct** - compares string and integer, returns False

eq2 = 5 == "3" # **Correct** - compares integer and string, returns False

eq3 = "5" == "3" # **Correct** - compares strings, returns False

eq4 = 5.0 == 3 # **Correct** - compares float and integer, returns False

eq5 = True == 3 # **Correct** - compares boolean and integer, returns False (True is equivalent to 1 in Python)

# Not Equal

neq = 5 != 3 # **Correct** - compares integers

neq1 = "5" != 3 # **Correct** - compares string and integer, returns True

neq2 = 5 != "3" # **Correct** - compares integer and string, returns True

neq3 = "5" != "3" # **Correct** - compares strings, returns True

neq4 = 5.0 != 3 # **Correct** - compares float and integer, returns True

neq5 = True != 3 # **Correct** - compares boolean and integer, returns True

neq6 = 5 != 4 # **Incorrect** - variable indentation is Incorrect (4 spaces or tab)

# Greater than

gt = 5 > 3 # **Correct** - compares integers

gt1 = "5" > 3 # **Incorrect** - TypeError, can't compare string and integer

gt2 = 5 > "3" # **Incorrect** - TypeError, can't compare integer and string

gt3 = "5" > "3" # **Correct** - compares strings, but note that it's lexicographical comparison

gt4 = 5.0 > 3 # **Correct** - compares float and integer

gt5 = True > 3 # **Correct** - compares boolean and integer, returns False (True is equivalent to 1 in Python)

# other similar comparison operators:

# Less than <

# Greater than or equal >=

# Less than or equal <=

# 4.Group Work – Truthy vs Falsy values and Logical Operators (3-4 people in a group)

7 minutes in a group #**Correct** or #**Incorrect** | #**Truthy** or #**Falsy** (if possible – why?)

4 minutes check against answers (one page without answers | 4 with answers to keep)

# **Truthy** vs **Falsy** values

print(bool(None))

print(bool(False))

print(bool(True))

print(bool(0))

print(bool(1))

print(bool(-1))

print(bool(0.1))

print(bool(0.0))

print(bool(""))

print(bool("Hello"))

print(bool([1, 2, 3]))

print(bool([]))

print(bool({}))

print(bool({"name": "John"}))

# And

and\_op = True and False

and\_op1 = "True" and False

and\_op2 = True and "False"

and\_op3 = "True" and "False"

and\_op4 = 1 and 0

and\_op5 = 1.0 and 0

# Or

or\_op = True or False

or\_op1 = "True" or False

or\_op2 = False or "False"

or\_op3 = "" or "False"

or\_op4 = 0 or 1

or\_op5 = 0.0 or 1.0

# Not

not\_op = not True

not\_op1 = not "True"

not\_op2 = not False

not\_op3 = not ""

not\_op4 = not 0

not\_op5 = not 0.0

# 4.Answers - Truthy vs Falsy values and Logical Operators

# **Truthy** vs **Falsy** values

print(bool(None)) # **Falsy** - False

print(bool(False)) # **Falsy** - False

print(bool(True)) # **Truthy** - True

print(bool(0)) # **Falsy** - False

print(bool(1)) # **Truthy** - True

print(bool(-1)) # **Truthy** - True

print(bool(0.1)) # **Truthy** - True

print(bool(0.0)) # **Falsy** - False

print(bool("")) # **Falsy** - False

print(bool("Hello")) # **Truthy** - True

print(bool([1, 2, 3])) # **Truthy** - True

print(bool([])) # **Falsy** - False

print(bool({})) # **Falsy** - False

print(bool({"name": "John"})) # **Truthy** - True

# And

and\_op = True and False # **Correct** - returns False

and\_op1 = "True" and False # **Correct** - returns False because 'and' operation with a **Falsy** value (False) returns the **Falsy** value

and\_op2 = True and "False" # **Correct** - returns "False" because 'and' operation with a **Truthy** value (True) returns the next value

and\_op3 = "True" and "False" # **Correct** - returns "False" because 'and' operation with a **Truthy** value ("True") returns the next value

and\_op4 = 1 and 0 # **Correct** - returns 0 because 'and' operation with a **Truthy** value (1) returns the next value

and\_op5 = 1.0 and 0 # **Correct** - returns 0 because 'and' operation with a **Truthy** value (1.0) returns the next value

# Or

or\_op = True or False # **Correct** - returns True

or\_op1 = "True" or False # **Correct** - returns "True" because 'or' operation with a **Truthy** value ("True") returns the **Truthy** value

or\_op2 = False or "False" # **Correct** - returns "False" because 'or' operation with a **Falsy** value (False) returns the next value

or\_op3 = "" or "False" # **Correct** - returns "False" because 'or' operation with a **Falsy** value ("") returns the next value

or\_op4 = 0 or 1 # **Correct** - returns 1 because 'or' operation with a **Falsy** value (0) returns the next value

or\_op5 = 0.0 or 1.0 # **Correct** - returns 1.0 because 'or' operation with a **Falsy** value (0.0) returns the next value

# Not

not\_op = not True # **Correct** - returns False

not\_op1 = not "True" # **Correct** - returns False because 'not' operation with a **Truthy** value ("True") returns False

not\_op2 = not False # **Correct** - returns True

not\_op3 = not "" # **Correct** - returns True because 'not' operation with a **Falsy** value ("") returns True

not\_op4 = not 0 # **Correct** - returns True because 'not' operation with a **Falsy** value (0) returns True

not\_op5 = not 0.0 # **Correct** - returns True because 'not' operation with a **Falsy** value (0.0) returns True