SKILLS CITY

Fair Access to Technology Futures

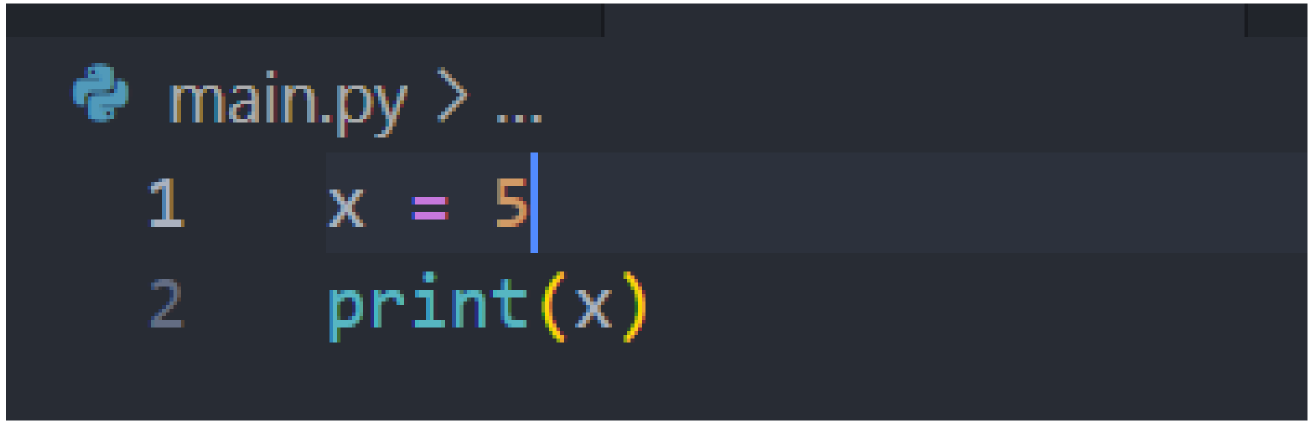
Variables & Data Types

IN4.0™ Group

Python is an interpreted language

SKILLS CITY
Fair Access to Technology Futures

This means that the code is executed from left to right, one line at a time, just like reading a page in a book.



```python
x = 5
print(x)
```

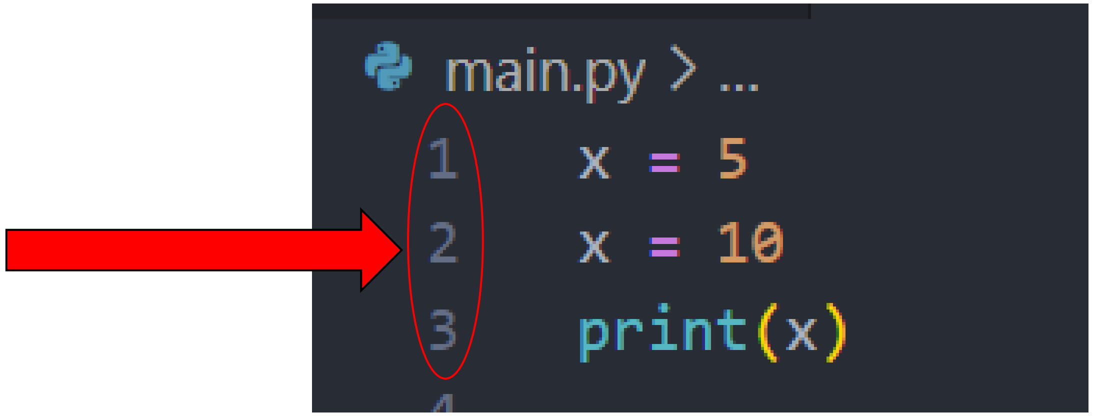
What will be the output of this code?
Answers in chat please!

Answer: 5

Python is an interpreted language

This means that the code is executed from left to right, one line at a time, just like reading a page in a book.

Line 2 re-assigns the value of x to 10


main.py > ...

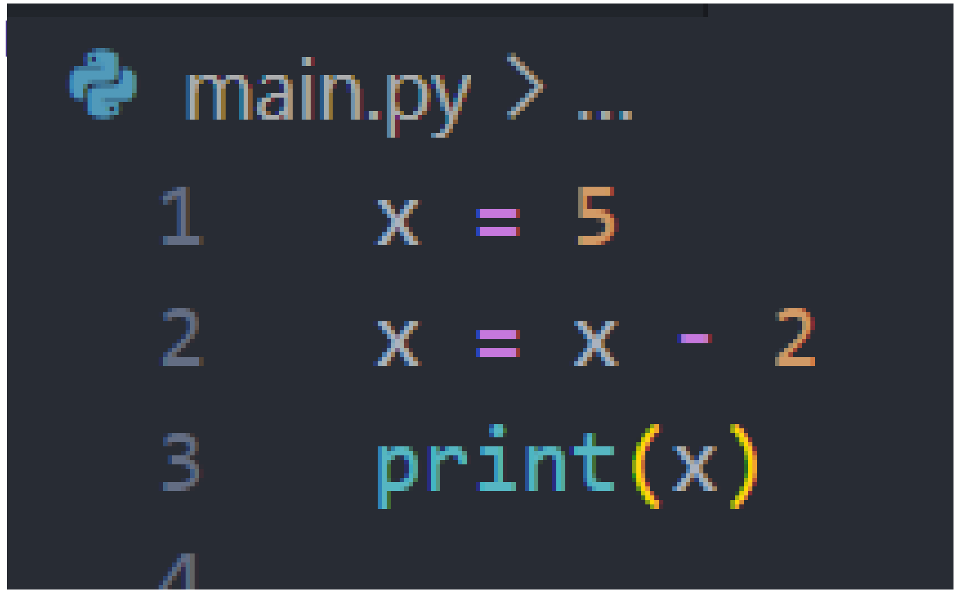
```
1    x = 5
2    x = 10
3    print(x)
```

What will be the output of this code?
Answers in chat please!

Answer: 10

Python is an interpreted language

This means that the code is executed from left to right, one line at a time, just like



main.py > ...

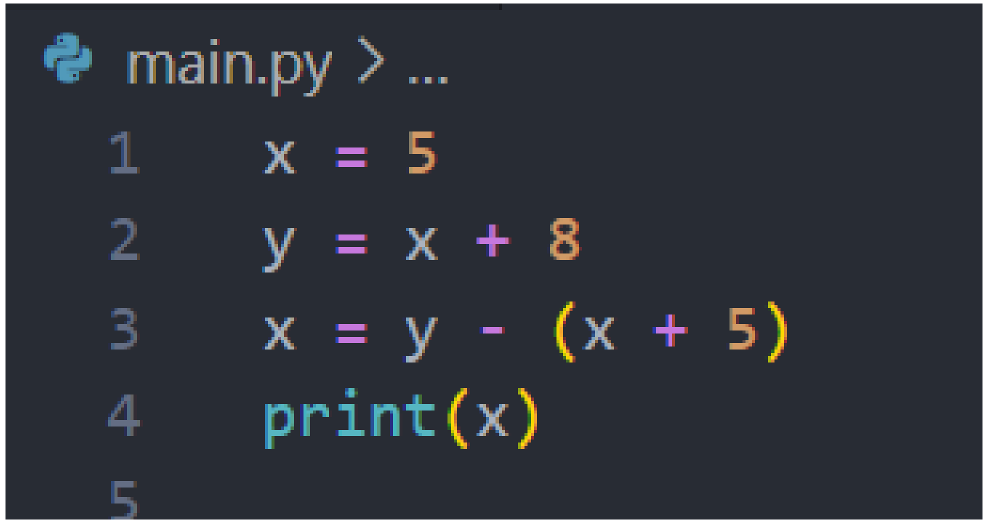
```python
x = 5
x = x - 2
print(x)
```

What will be the output of this code?
Answers in chat please!

Answer: 3

Python is an interpreted language

BIDMAS still applies to code!



main.py > ...

```python
x = 5
y = x + 8
x = y - (x + 5)
print(x)

```

- x = 5
- y = x + 8
- y = 5 + 8
- y = 13
- x = y - (x + 5)
- x = 13 - (5 + 5)
- x = 13 - 10
- x = 3

What will be the output of this code?
Answers in chat please!

Answer:

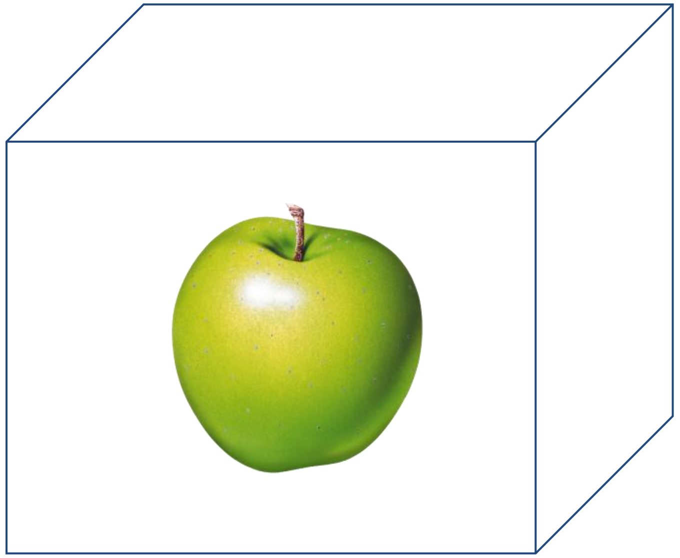
Variables

Variables are containers for storing data values.

Variables

A variable can simply be viewed as a box that contains something.

This box contains an apple. If this was a variable, we could write:



box = “apple”

This is the computer version of putting an apple into a box. Knowing where it is, so we can use it later.

Python Variables

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)

Variables

SKILLS CITY
Fair Access to Technology Futures

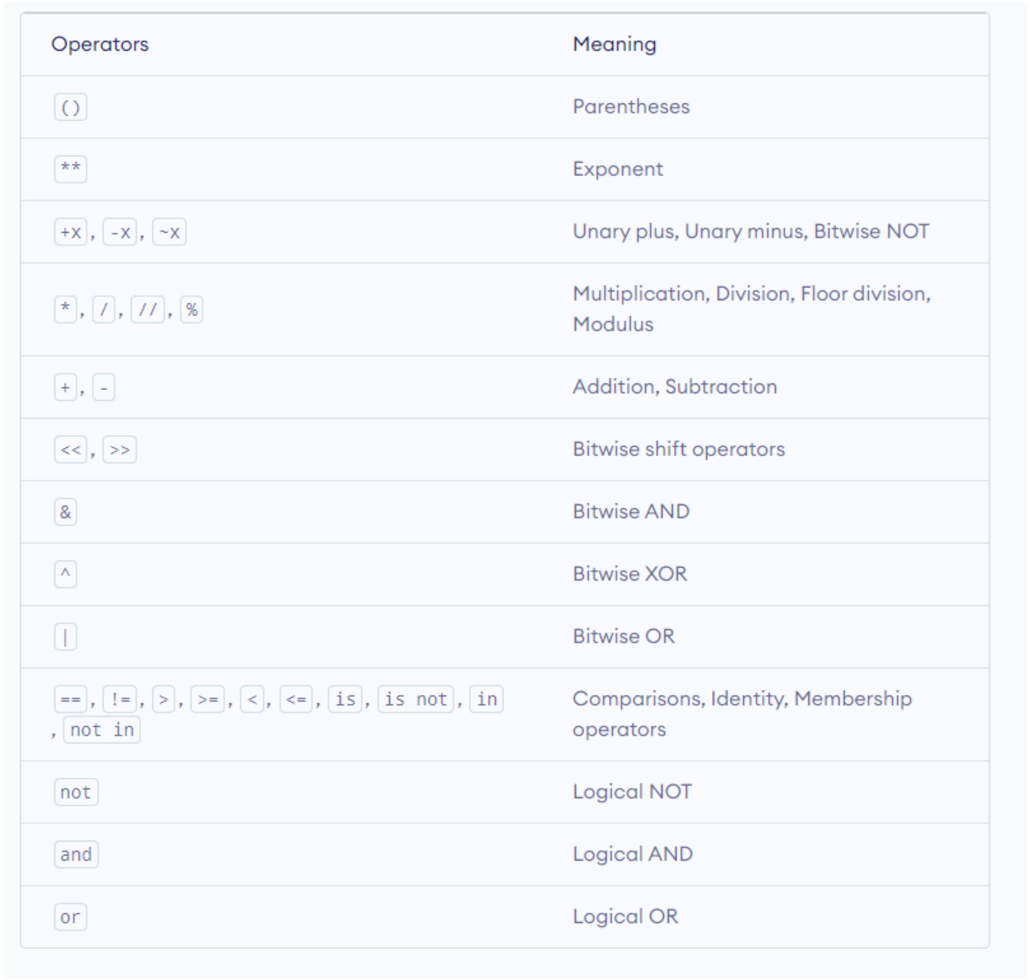
```
x = 5
y = "John“
z = ‘c’
print(x)
print(y)
```

Variables – Assign Multiple

SKILLS CITY
Fair Access to Technology Futures

```
x, y, z = True, "Banana", 5
print(x)
print(y)
print(z)
```

Variables – Operator Precedence



Operators	Meaning
`()`	Parentheses
`**`	Exponent
`+x`, `-x`, `~x`	Unary plus, Unary minus, Bitwise NOT
`*`, `/`, `//`, `%`	Multiplication, Division, Floor division, Modulus
`+`, `-`	Addition, Subtraction
`<<`, `>>`	Bitwise shift operators
`&`	Bitwise AND
`^`	Bitwise XOR
`|`	Bitwise OR
`==`, `!=`, `>`, `>=`, `<`, `<=`, `is`, `is not`, `in`, `not in`	Comparisons, Identity, Membership operators
`not`	Logical NOT
`and`	Logical AND
`or`	Logical OR

Data Types

In programming, data type is an important concept.

- Determines what sort of data will be handled by the computer program
- It is important to consider how you want your program to handle data.
- Variables can store data of different types, and different types can do different things.

(e.g integers can be used for mathematical operations, strings cannot!)

Data Types

string - used to store characters, which can be text and numbers. represented with “quotation marks” or ‘single quotes’

integer - used to store whole numbers

float - used to store decimal point numbers

boolean - used to store two possible values (True / False)

Data Types

SKILLS CITY
Fair Access to Technology Futures

string - “this is a string”   “th15 15 4ls0 4 str1ng!”

integer - 123

float - 1.23

boolean - True / False

Dynamically Typed Language

Python is dynamically typed, which means that variable types are determined and checked at runtime rather than during compilation. In dynamically typed languages like Python, you don't need to explicitly declare the variable type before using it.

Data Types

SKILLS CITY
Fair Access to Technology Futures

Integers are devoid of the fractional part. It is a whole number.

Python rules: does not accept any other characters other than underscore in between numbers

2_22_222_000

For the math lovers, you might want to know that Python works with octal and hexadecimal numbers! I will definitely not go through this on this course =)

Data Types

SKILLS CITY
Fair Access to Technology Futures

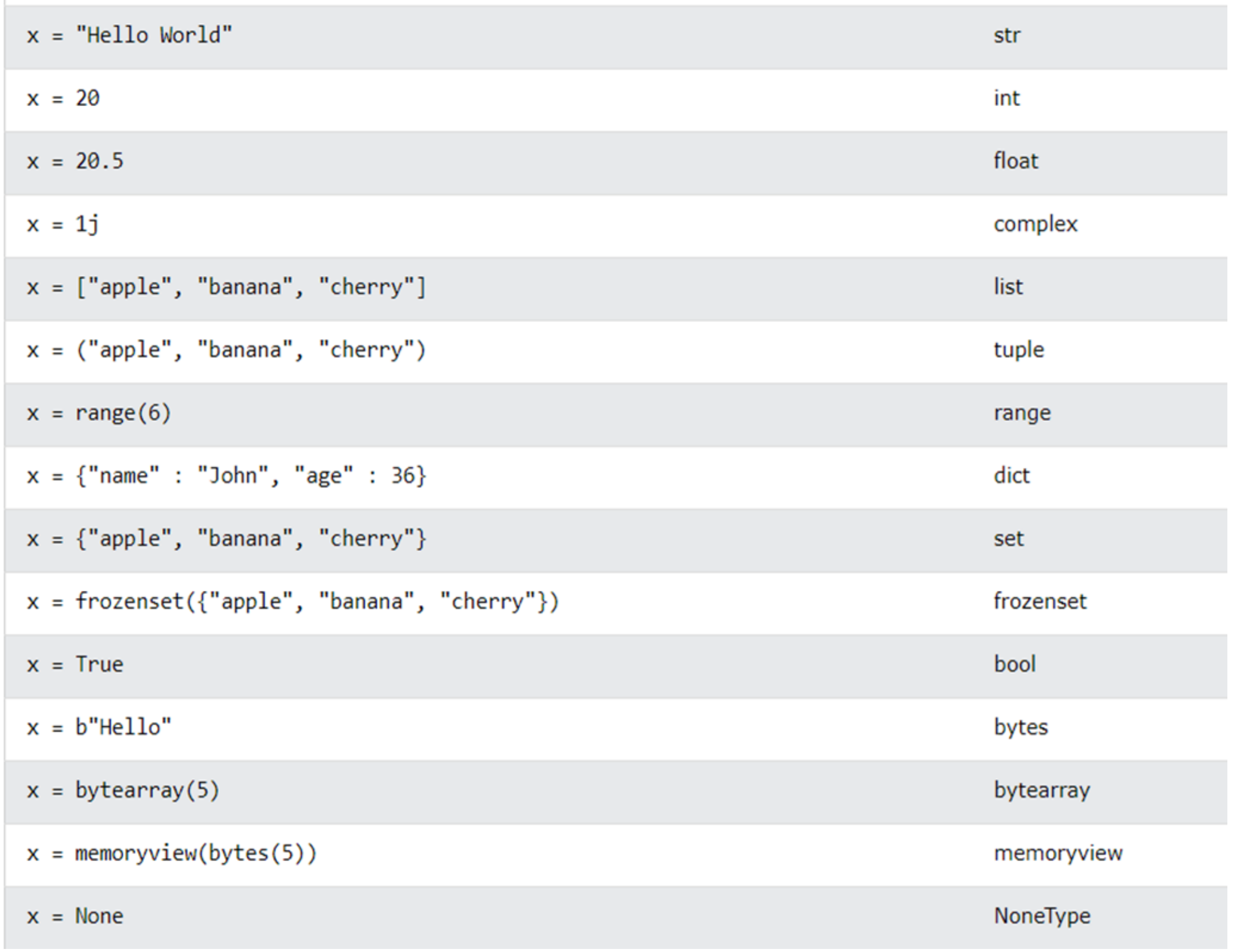
Float, or "floating point number" is a number, positive or negative, containing one or more decimals.

```
x = 1.10
y = 1.0
z = -35.59

print(type(x))
print(type(y))
print(type(z))
```

Data Types

SKILLS CITY
Fair Access to Technology Futures



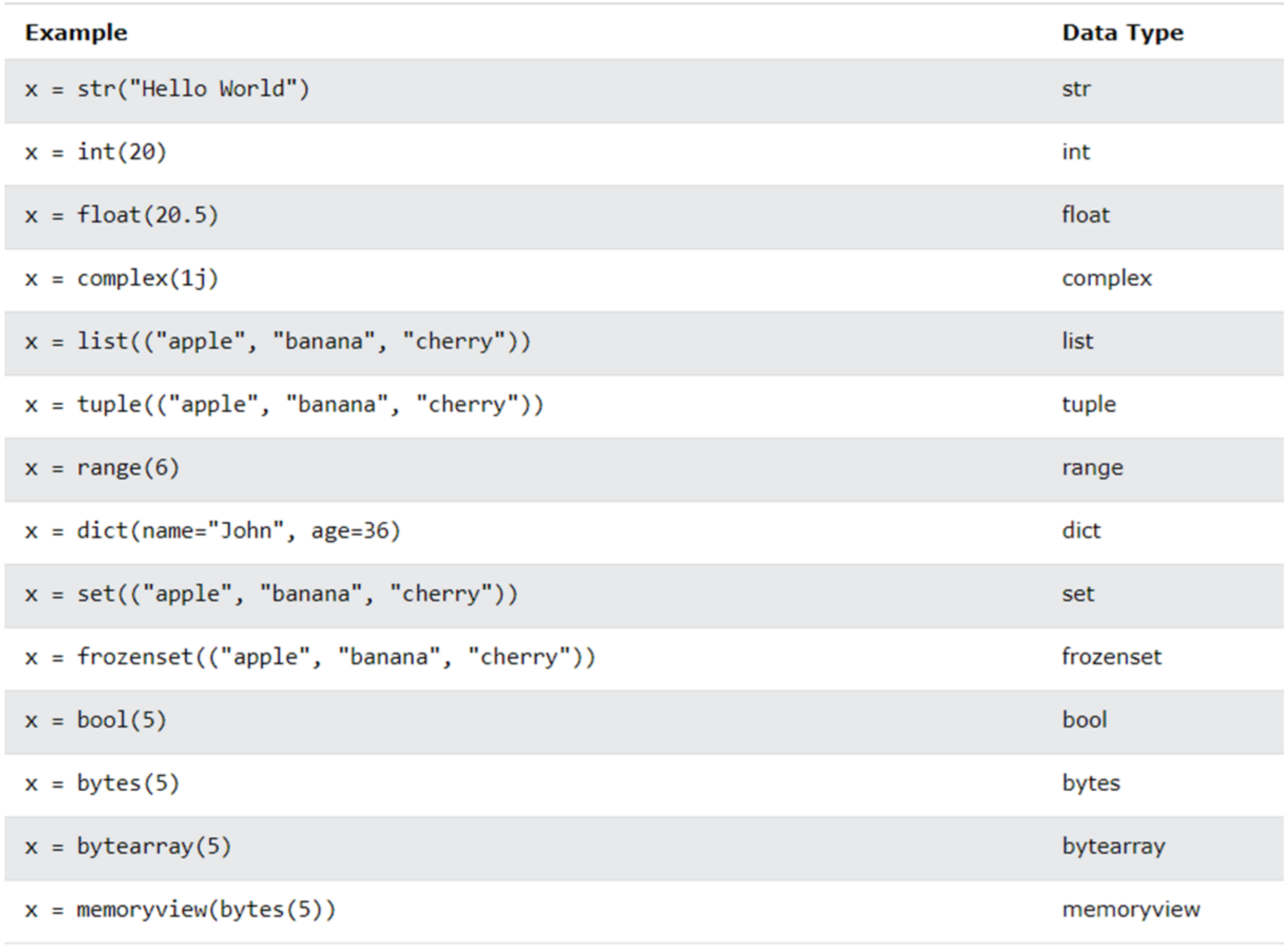
`x = "Hello World"`	str
`x = 20`	int
`x = 20.5`	float
`x = 1j`	complex
`x = ["apple", "banana", "cherry"]`	list
`x = ("apple", "banana", "cherry")`	tuple
`x = range(6)`	range
`x = {"name" : "John", "age" : 36}`	dict
`x = {"apple", "banana", "cherry"}`	set
`x = frozenset({"apple", "banana", "cherry"})`	frozenset
`x = True`	bool
`x = b"Hello"`	bytes
`x = bytearray(5)`	bytearray
`x = memoryview(bytes(5))`	memoryview
`x = None`	NoneType

In Python, the data type is set by default when you assign a value to a variable

In other words, Python knows what you're talking about, without having to tell it; which value is a string, which value is an integer, etc..

Data Types

SKILLS CITY
Fair Access to Technology Futures



Example	Data Type
`x = str("Hello World")`	str
`x = int(20)`	int
`x = float(20.5)`	float
`x = complex(1j)`	complex
`x = list(("apple", "banana", "cherry"))`	list
`x = tuple(("apple", "banana", "cherry"))`	tuple
`x = range(6)`	range
`x = dict(name="John", age=36)`	dict
`x = set(("apple", "banana", "cherry"))`	set
`x = frozenset(("apple", "banana", "cherry"))`	frozenset
`x = bool(5)`	bool
`x = bytes(5)`	bytes
`x = bytearray(5)`	bytearray
`x = memoryview(bytes(5))`	memoryview

You can also set a **SPECIFIC** **data type** if you want to

If you want to specify the data type, you can use the constructor functions :

x = int(20)

This is particularly useful for conditional statements.

Data Types

SKILLS CITY
Fair Access to Technology Futures

Python has the following data types built-in by default, in these categories:

Don’t worry you don’t need to know all of them! What you should know is how to find this information using Python documentation.

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview
None Type:	NoneType

Task Breakdown

SKILLS CITY
Fair Access to Technology Futures

You can get the data type of any object by using the **type()** function:

Type the following code to output the datatype of the variable **“age”**

```
age = 30
print(type(age))
```

Task Breakdown

Correct the code below so that Python can perform an addition operation between the two numbers:

```
x = 5
y = 10
print(x + y)  # 15
```

Task Breakdown

SKILLS CITY
Fair Access to Technology Futures

```
y = int(2.8) # y will be
y = float(2) # y will be
y = str(2) # y will be
```

Task Breakdown

SKILLS CITY
Fair Access to Technology Futures

```python
a = "Hello"
b = "World"
c = a + b
print(c)
```

Task Breakdown

SKILLS CITY
Fair Access to Technology Futures

```
age = 36
txt = "My name is John, I am " + age
print(txt)
```

Task Breakdown

SKILLS CITY
Fair Access to Technology Futures

```
age = 36
txt = "My name is John, and I
am {}"
print(txt.format(age))
```

Task Breakdown

SKILLS CITY
Fair Access to Technology Futures

```
print(10 > 9)
print(10 == 9)
print(10 < 9)
```

Task Breakdown

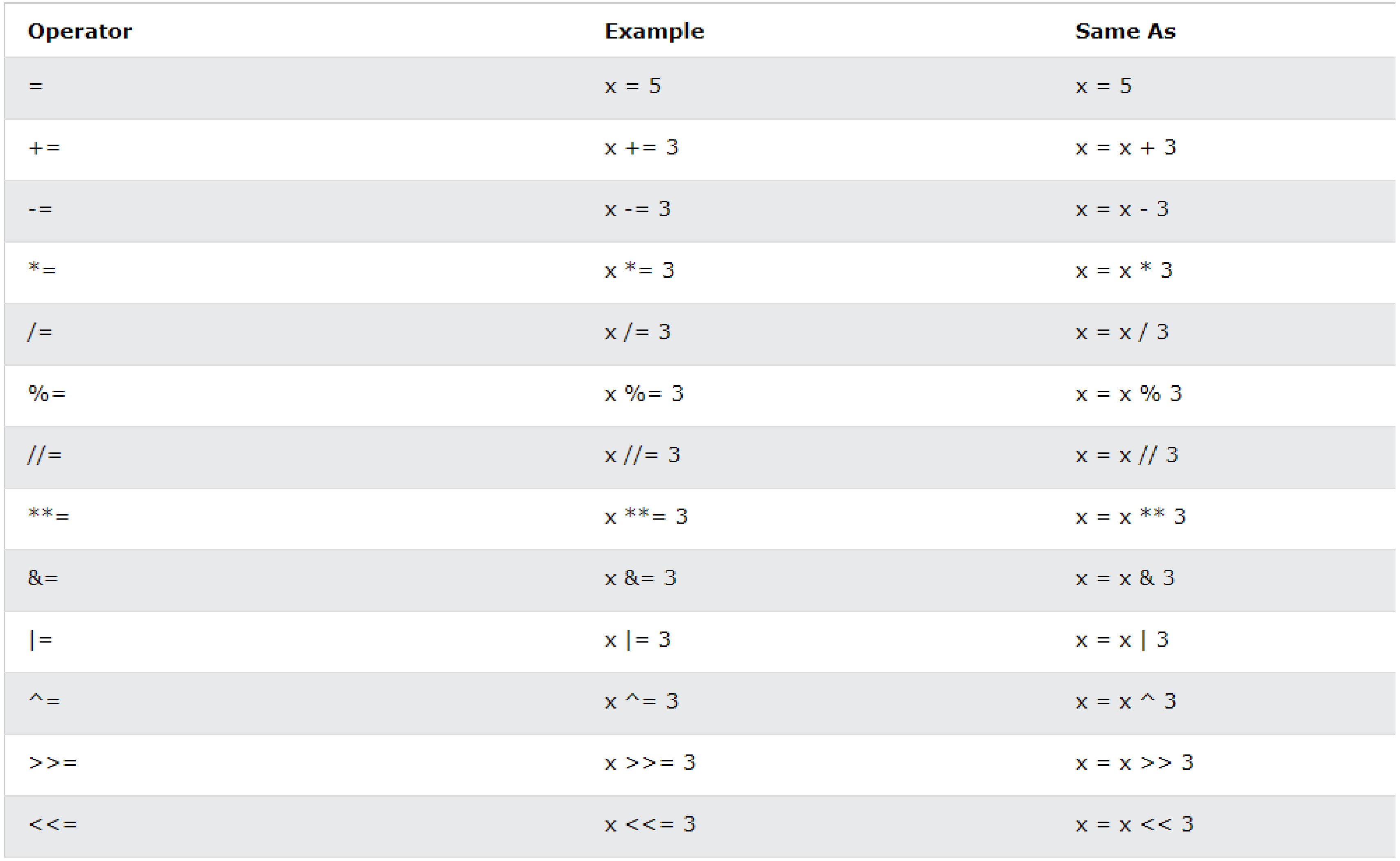
SKILLS CITY
Fair Access to Technology Futures

```
x = "Hello"
y = 15

print(bool(x))
print(bool(y))
```

Task Breakdown - Operators

SKILLS CITY
Fair Access to Technology Futures



Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3