

Variables, Expressions, and Statements

Chapter 2



Python for Everybody
www.py4e.com



Constants

- **Fixed values** such as numbers, letters, and strings, are called “**constants**” because their value does not change
- Numeric **constants** are as you expect
- String **constants** use single quotes (') or double quotes (")

```
>>> print(123)
```

```
123
```

```
>>> print(98.6)
```

```
98.6
```

```
>>> print('Hello world')
```

```
Hello world
```


Reserved Words

- You cannot use **reserved words** as variable names / identifiers

False	class	return	is	finally
None	if	for	lambda	continue
True	def	from	while	nonlocal
and	del	global	not	with
as	elif	try	or	yield
assert	else	import	pass	
break	except	in	raise	

Variables

- A **variable** is a named place in the memory where a programmer can store data and later retrieve the data using the **variable** “name”
- Programmers get to choose the names of the **variables**
- You can change the contents of a **variable** in a later statement

x = 12.2

y = 14

x 12.2

y 14

Variables

- A **variable** is a named place in the memory where a programmer can store data and later retrieve the data using the **variable** “name”
- Programmers get to choose the names of the **variables**
- You can change the contents of a **variable** in a later statement

x = 12.2

y = 14

x = 100

x ~~12.2~~ 100

y 14

Python Variable Name Rules

Must start with a letter or underscore _

Must consist of letters, numbers, and underscores

Case Sensitive

Good: spam eggs spam23 _speed

Bad: 23spam #sign var.12

Different: spam Spam SPAM

Sentences or Lines

<code>x</code>	<code>=</code>	<code>2</code>	←	Assignment statement		
<code>x</code>	<code>=</code>	<code>x</code>	<code>+</code>	<code>2</code>	←	Assignment with expression
<code>print</code>	<code>(</code>	<code>x</code>	<code>)</code>	←	Print statement	

Variable

Operator

Constant

Function

Mnemonic Variable Names

- Since we programmers are given a choice in how we choose our variable names, there is a bit of “best practice”
- We name variables to help us remember what we intend to store in them (“**mnemonic**” = “memory aid”)
- This can confuse beginning students because well-named variables often “sound” so good that they must be keywords

<http://en.wikipedia.org/wiki/Mnemonic>


```
x1q3z9ocd = 35.0  
x1q3z9afd = 12.50  
x1q3p9afd = x1q3z9ocd * x1q3z9afd  
print(x1q3p9afd)
```

What is this bit of
code doing?


```
x1q3z9ocd = 35.0  
x1q3z9afd = 12.50  
x1q3p9afd = x1q3z9ocd * x1q3z9afd  
print(x1q3p9afd)
```

```
a = 35.0  
b = 12.50  
c = a * b  
print(c)
```

What are these
bits of code
doing?


```
x1q3z9ocd = 35.0  
x1q3z9afd = 12.50  
x1q3p9afd = x1q3z9ocd * x1q3z9afd  
print(x1q3p9afd)
```

```
a = 35.0  
b = 12.50  
c = a * b  
print(c)
```

What are these
bits of code
doing?

```
hours = 35.0  
rate = 12.50  
pay = hours * rate  
print(pay)
```

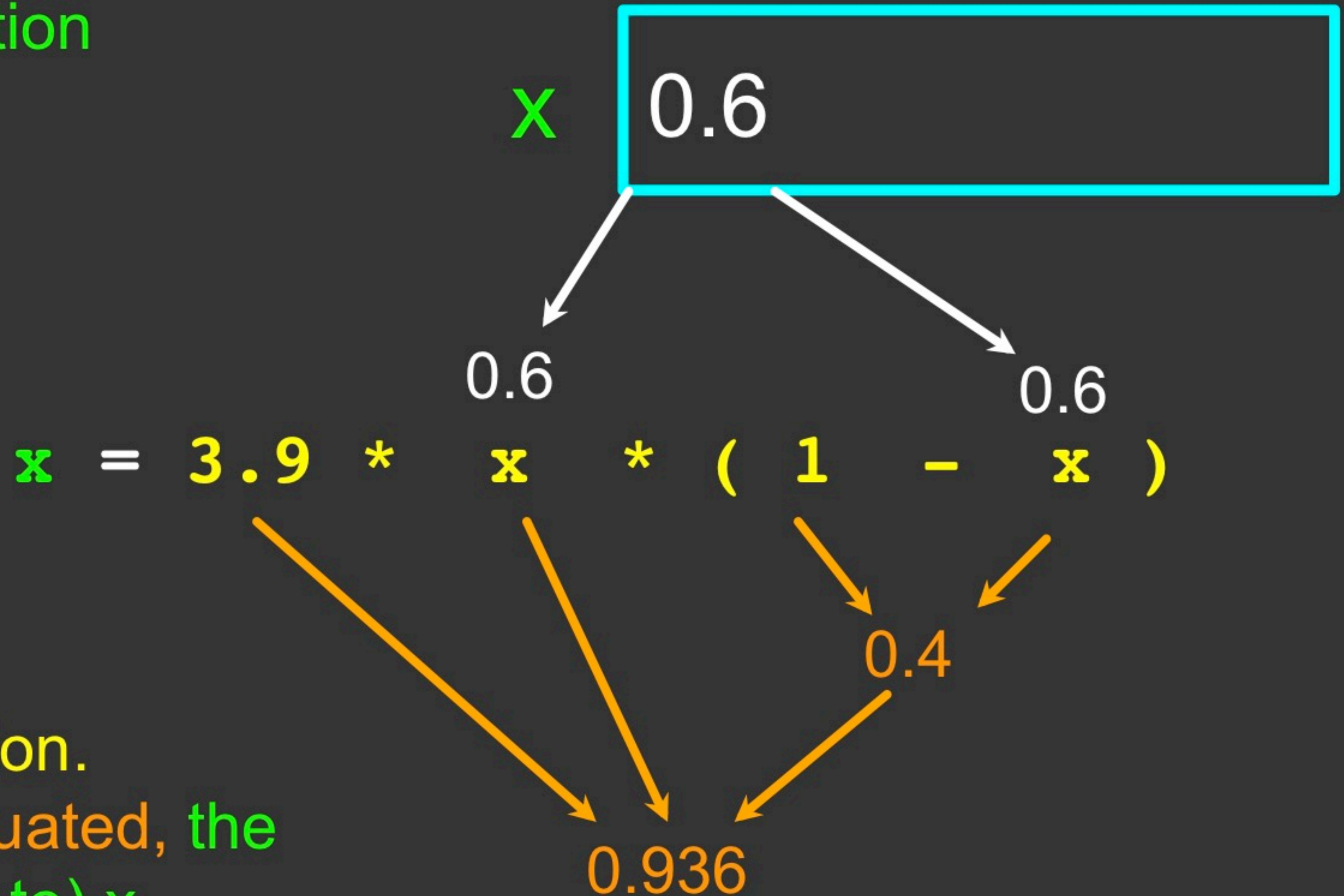

Assignment Statements

We assign a value to a variable using the assignment statement (=)

An assignment statement consists of an **expression on the right-hand side** and a **variable** to store the result

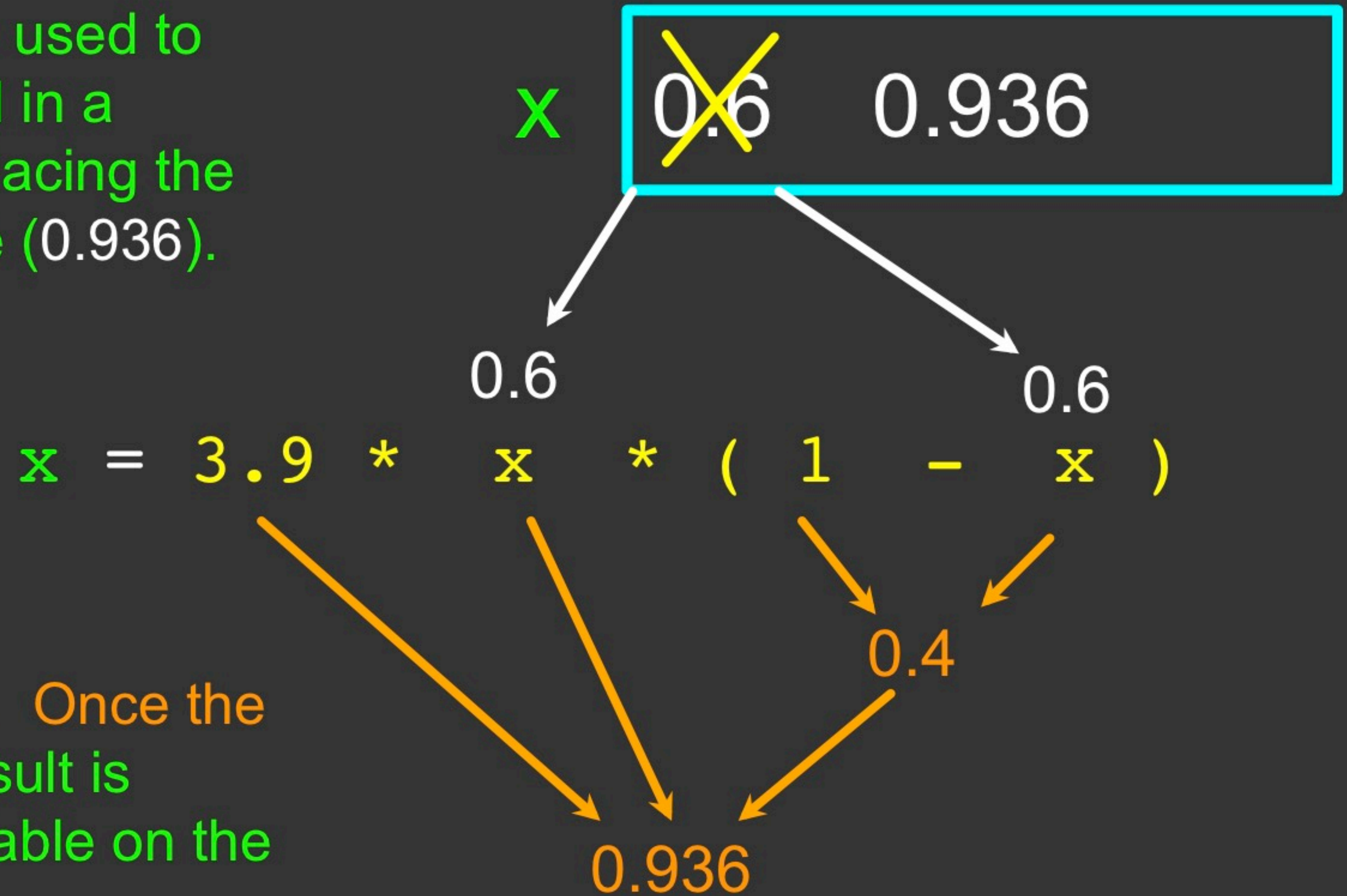
```
x = 3.9 * x * ( 1 - x )
```


A variable is a memory location used to store a value (0.6)



The right side is an expression.
Once the expression is evaluated, the result is placed in (assigned to) `x`.

A variable is a memory location used to store a value. The value stored in a variable can be updated by replacing the old value (0.6) with a new value (0.936).



The right side is an expression. Once the expression is evaluated, the result is placed in (assigned to) the variable on the left side (i.e., `x`).

Expressions