ariables, Expressions, a Statements

Chapter 2

Python for Everybody

www.py4e.com

Constants

values such as numbers, letters, and strings, are
tants" because their value does not change

eric constants are as you expect

g constants use single quotes (')
uble quotes (")

```
>>> print(123)
123
>>> print(98.6)
98.6
>>> print('Hello world')
Hello world
```

Reserved Words

ot use reserved words as variable names / identifie

```
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

le is a named place in the memory where a programmer
d later retrieve the data using the variable “name”

mers get to choose the names of the variables

change the contents of a variable in a later statement

```
= 12.2
= 14
```

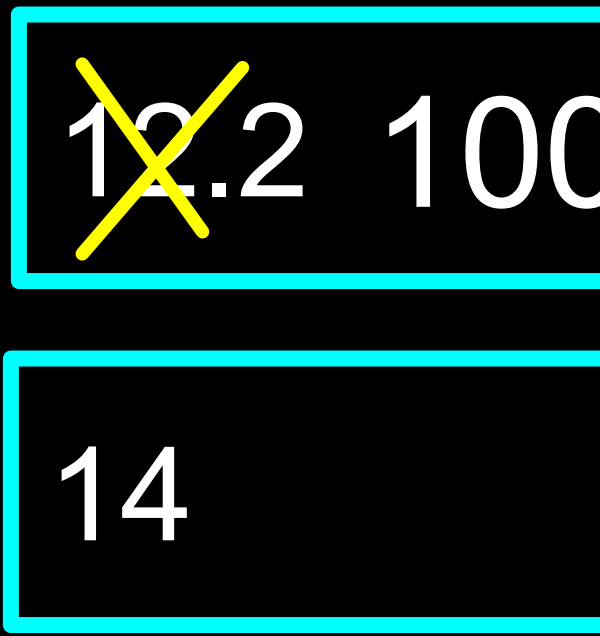
Variables

le is a named place in the memory where a programmer
d later retrieve the data using the variable “name”

mers get to choose the names of the variables

change the contents of a variable in a later statement

```
= 12.2
= 14
= 100
```



thon Variable Name Ru

start with a letter or underscore _

consist of letters, numbers, and underscores

Sensitive

```
od:      spam    eggs   spam23     _speed
l:      23spam      #sign  var.12
fferent:     spam   Spam   SPAM
```

nemonic Variable Name

we programmers are given a choice in how we ch
ble names, there is a bit of “best practice”

ame variables to help us remember what we intend
m (“mnemonic” = “memory aid”)

can confuse beginning students because well-nam
bles often “sound” so good that they must be keyw

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

```
d = 35.0
d = 12.50
d = x1q3z9ocd * x1q3z9afd
q3p9afd)
```

s this bit of
e doing?

```
d = 35.0                                    a = 35.
d = 12.50                                   b = 12.
d = x1q3z9ocd * x1q3z9afd                   c = a *
q3p9afd)                                    print(c
```

re these bits
de doing?

```
d = 35.0
d = 12.50
d = x1q3z9ocd * x1q3z9afd
q3p9afd)
```

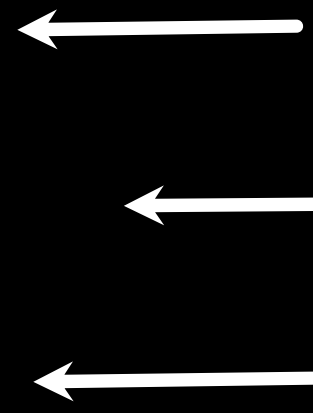
```
a = 35.
b = 12.
c = a *
print(c
```

re these bits
de doing?

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

Sentences or Lines

```
2
x + 2
t(x)
```



Assignment statem
Assignment with ex
Print statement

le Operator Constant Func

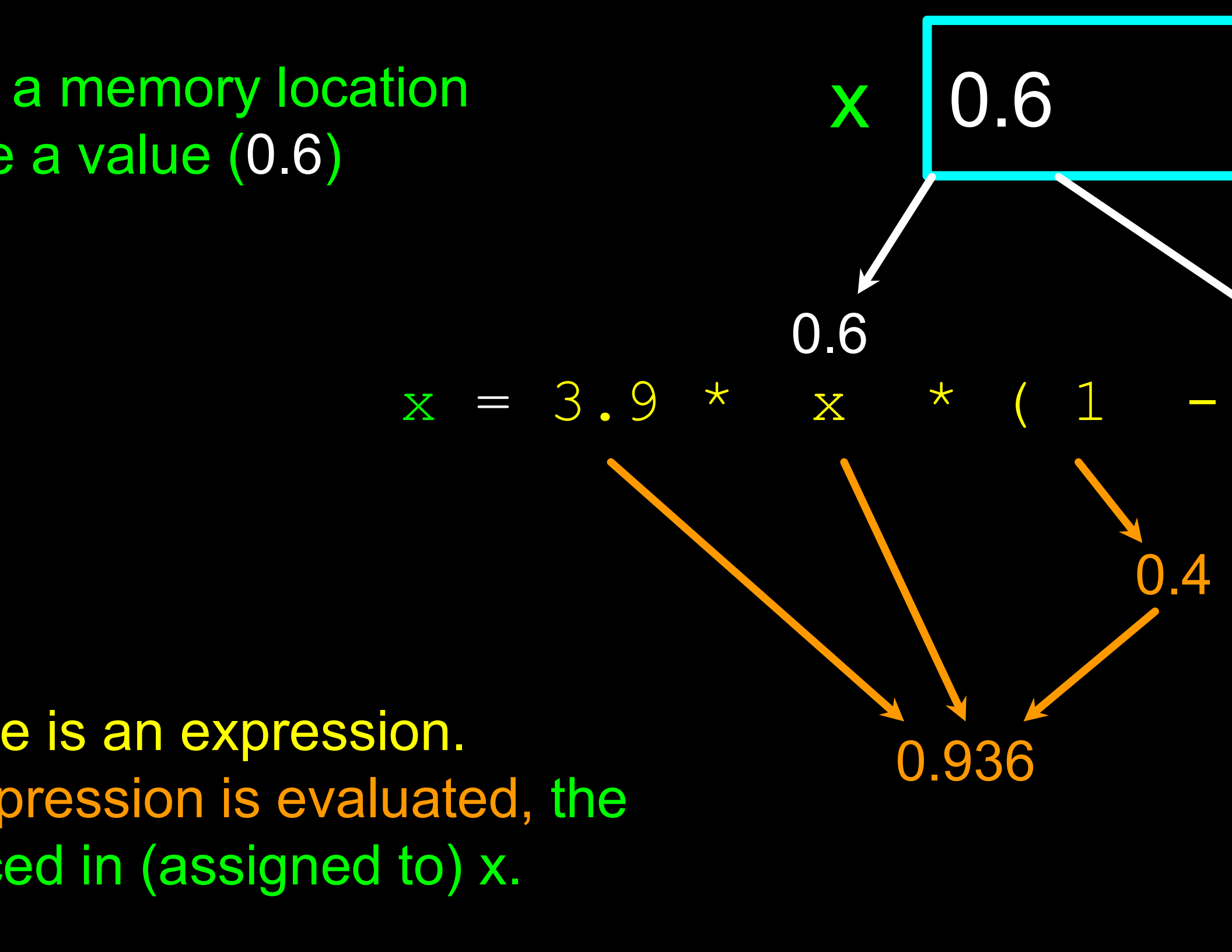
Assignment Statements

gn a value to a variable using the assignment state

nment statement consists of an expression on the
d side and a variable to store the result

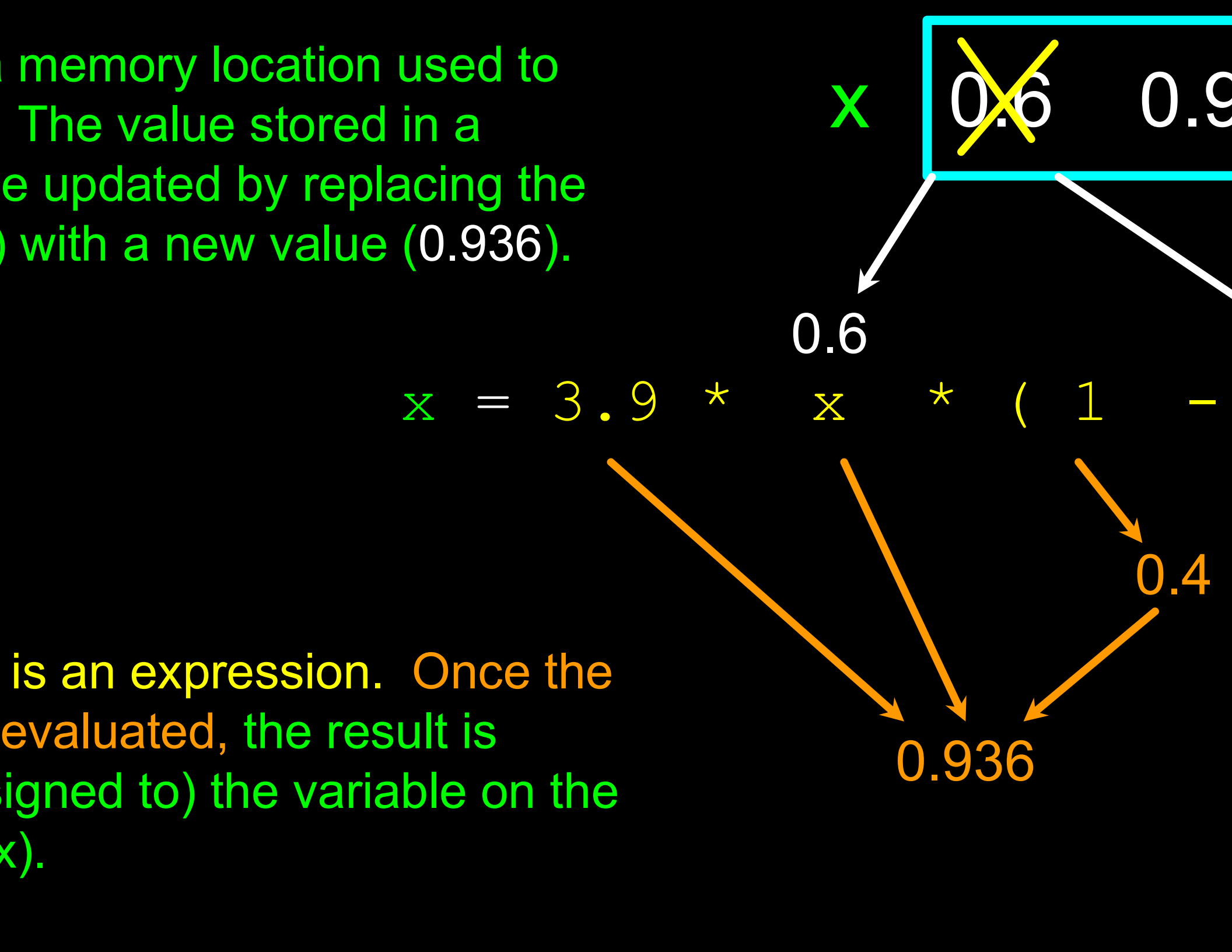
```
x = 3.9 * x * ( 1 - x )
```

a memory location
e a value (0.6)



e is an expression.
pression is evaluated, the
ced in (assigned to) x.

memory location used to
The value stored in a
e updated by replacing the
) with a new value (0.936).



is an expression. Once the
evaluated, the result is
igned to) the variable on the
x).

Expressions…

Numeric Expressions

se of the lack of mathematical
s on computer keyboards - we
mputer-speak" to express the
math operations

k is multiplication

entiation (raise to a power) looks
t than in math

Operator	
+	
-	S
*	M
/	
**	
%	

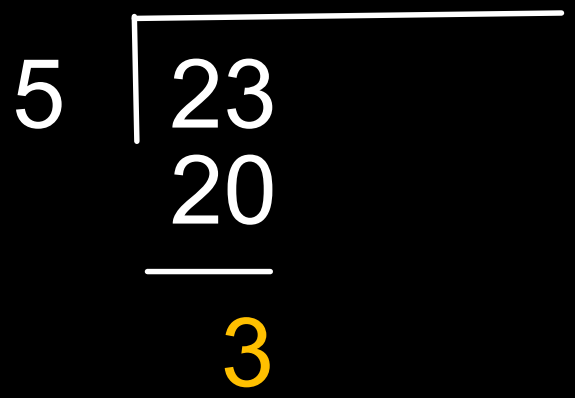
Numeric Expressions

```
 = 2
 = xx + 2
int(xx)

 = 440 * 12
int(yy)

 = yy / 1000
int(zz)
```

```
>>> jj = 23
>>> kk = jj % 5
>>> print(kk)
3
>>> print(4 ** 3)
64
```



Operato
+
-
*
/
**
%

Order of Evaluation

ve string operators together - Python must know w
rst

called “operator precedence”

operator “takes precedence” over the others?

```
x = 1 + 2 * 3 - 4 / 5 ** 6
```

perator Precedence Rul

cedence rule to lowest precedence rule:

heses are always respected

entiation (raise to a power)

ication, Division, and Remainder

on and Subtraction

right

Pare

P

Multi

Ac

Left

```
= 1 + 2 ** 3 / 4 * 5
int(x)
```

```
1 + 2 ** 3 /
```

```
1 + 8 / 4
```

```
1 + 2 *
```

```
1 + 10
```

```
11
```

Parenthesis
Power
Multiplication
Addition
Left to Right

erator Precedence

Paren
Pow
Multipl
Add
Left to

nber the rules top to bottom

vriting code - use parentheses

vriting code - keep mathematical expressions simp
y are easy to understand

ong series of mathematical operations up to make
ear

What Does "Type" Mean

on variables, literals, and
nts have a "type"

knows the difference between
ger number and a string

mple "+" means "addition" if
ing is a number and
enate" if something is a string

```
>>> ddd = 1 + 4
>>> print(ddd)
5
>>> eee = 'hello
>>> print(eee)
hello there
```

concatenate = pu

Type Matters

knows what "type"
ing is

perations are
ed

nnot "add 1" to a string

ask Python what type
ing is by using the
unction

```
>>> eee = 'hello ' + '
>>> eee = eee + 1
Traceback (most recent
File "<stdin>", line 1
<module>TypeError: Can
'int' object to str im
>>> type(eee)
<class'str'>
>>> type('hello')
<class'str'>
>>> type(1)
<class'int'>
>>>
```

everal Types of Number

rs have two main types

ers are whole numbers:
2, 0, 1, 100, 401233

ng Point Numbers have
parts: -2.5 , 0.0, 98.6, 14.0

are other number types - they
ations on float and integer

```
>>> xx = 1
>>> type (
<class 'in
>>> temp =
>>> type(t
<class'flo
>>> type(1
<class 'in
>>> type(1
<class'flo
>>>
```

Conversions

ou put an integer and
point in an
sion, the integer is
y converted to a float

n control this with the
functions int() and

```
>>> print(float(9
199.0
>>> i = 42
>>> type(i)
<class'int'>
>>> f = float(i)
>>> print(f)
42.0
>>> type(f)
<class'float'>
>>>
```

Integer Division

ivision produces a floating
ult

```
>>> print(10 / 2
5.0
>>> print(9 / 2)
4.5
>>> print(99 / 1
0.99
>>> print(10.0 /
5.0
>>> print(99.0 /
0.99
```

different in Python 2.x

String nversions

n also use int() and
o convert between
and integers

l get an error if the string
ot contain numeric
ers

```
>>> sval = '123'
>>> type(sval)
<class 'str'>
>>> print(sval + 1)
Traceback (most recent c
File "<stdin>", line 1,
TypeError: Can't convert
to str implicitly
>>> ival = int(sval)
>>> type(ival)
<class 'int'>
>>> print(ival + 1)
124
>>> nsv = 'hello bob'
>>> niv = int(nsv)
Traceback (most recent c
File "<stdin>", line 1,
ValueError: invalid lite
with base 10: 'x'
```

User Input

an instruct Python to
e and read data from
ser using the input()
on

nput() function
ns a string

```
nam = input('Who ar
print('Welcome', na
```

```
Who are you? Ch
Welcome Chuck
```

verting User Input

want to read a number
the user, we must
ert it from a string to a
ber using a type
ersion function

```
inp = input('Europe
usf = int(inp) + 1
print('US floor', us
```

we will deal with bad
data

Europe floor?
US floor 1

Comments in Python

g after a # is ignored by Python

mment?

ribe what is going to happen in a sequence of cod

ment who wrote the code or other ancillary inform

off a line of code - perhaps temporarily

```
# Get the name of the file and open it
name = input('Enter file:')
handle = open(name, 'r')

# Count word frequency
counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1

# Find the most common word
bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

# All done
print(bigword, bigcount)
```

Summary

- erved words
- ables (mnemonic)
- ators
- ator precedence
- Integer Division
- Conversion betwe
- User input
- Comments (#)

Write a program to prompt the user for hours and rate per hour to compute gross pay.

```
Enter Hours: 35
Enter Rate: 2.75

Pay: 96.25
```

Acknowledgements / Contributions

...

t: Charles Severance, University of Michigan
ion

tributors and Translators here