Cyber Security

Jenjira Jaimunk

Today….

- Assignment operator
- Mathematical operators
- Boolean expressions
- Relational operators
- Random numbers

Assignment operator

- = is the assignment operator
- example:

  program code:

```
# declare variable and assign it a value
x = 19
```

computer's memory:

x → $\boxed{19}$

- read "x is assigned the value 19" or "x gets 19"

Mathematical operators

- the basic mathematical operators built into Python are:

+	addition	+	unary plus (positive sign)
−	subtraction	−	unary minus (negative sign)
*	multiplication	**	power
/	division	%	modulo

- example:

```
x = −5
y =  x  *7
y =  y + 3
x =  x  *−2
y =  x  / 19
print x
print y
```

- what is output?
- *note that modulo means "remainder after integer division"*

Boolean expressions

- Boolean values: `True` and `False`
- the logical operators are **not**, **and**, **or**
- defined by "`truth`" tables:

A	B	not A	A and B	A or B
True	True	False	True	True
True	False		False	True
False	True	True	False	True
False	False		False	False

- example:

```
x = True
y = False
print(x and y)
print(not x)
print (x or not y)
```

- what is output?

Relational operators

- relational operators are used to compare two values against each other
- which can result in equality or some type of inequality
- operators:

==	equality	! =	inequality
>	greater than	>=	greater than or equal to
<	less than	<=	Less than or equal to

- example:

```
x = −5
y = 7
print ( x < y )
print (  x == y )
print (  x >= y )
```

- what is the output?

Random numbers (1)

- the random module is part of the Python standard library
- it can be used to generate pseudo-random numbers
- but it is not automatically built in, so in order to use the contents of the module, you need to "import" it, using a statement like this:

  ```
  import random
  ```

- the function random.random() will return a random number (float) in the half-closed (or half-open) interval [0.0,1.0)

  **remember:

  []indicate the endpoints of a *closed* interval (including the endpoint)

  ( ) indicate the endpoints of an *open* interval (not including the endpoint)

→ which means that we will get a random number *n*, such that

$0 <= n < 1$

Random numbers (2)

- the function random.randint(a,b) will return a random integer in the interval [*a*, *b*]
- this is the *closed* interval between *a* and *b*

→ which means that we will get a random number *n*, such that $a <= n <= b$

- example:

```
import random
i = random.randint(1,10)
print (’i=’ + str(i))
```

Questions and Answers