THINK PYTHON CH 2

THINK PYTHON CH 2 "Variables, expressions, and statements"

2.1 VALUES & TYPES

'Value' - basic thing program works with, like a letter or number

'Types' of values (datatype)

↳ 'string'

- contains a 'string' of letters
- enclosed in 'quotations'

* If you are not sure what type a value has, the interpreter can tell you

```
>>> type ('Hello World!')
<type 'str'>
>>> type (17)
<type 'int'>
```

↳ 'integers'

- numbers
- no quotes

↳ floating point

- numbers with a decimal point (can omit zero)

```
>>> type (3.2)
<type 'float'>
```

* BE CAREFUL

```
>>> 1,000,000
```

is interpreted as:

(1, 0, 0)

semantic error!
code runs, no bugs, but it doesn't do the right thing

2.2 VARIABLES

'Variable' - a name that refers to a value

'Assignment Statement' - creates new variables and gives them valu

```
>>> message = 'And now for something completely different'
>>> n = 17
>>> pi = 3.14159265358979323
```

↳ this makes three assignments

'State Diagram' - represent variables on paper

n ⟶ 17

* Type of variable is the type of value it refers to. EX:

```
>>> type (message)
<type 'str'>
```

2.3 VARIABLE NAME & KEYWORDS

- names are meaningful
  - ↳ tells us what variable is used for
- can contain both letters & numbers
- must start with letter (preferably lower case)

2.4 OPERATORS & OPERANDS

- PYTHON has 31 keywords these can't be used as variable n

PYTHON 3:

and	del	global	not	while
as	elif	if	or	with
assert	else	import	pass	yield
break	except	in	print	
class	finally	is	raise	
continue	for	lambda	return	
def	from	nonlocal	try	

'Operators' - special symbols, represent computation like addition and multiplication

'Operands' - values the operator is applied to

** is exponentiation

```
>>> minute = 59
>>> minute / 60
0
```

*??? * ↳ Floor division (integers) (FLOOR)

- If either of the operands are floating point numbers Python will perform floating point division.

```
>>> minute/60.0
0.9833333333333333328
```

2.5 EXPRESSIONS & STATEMENTS

'Expression' - combination of values, variables, and operators

'statement' - unit of code Python interpreter can execute.

*Expression has a value a statement does not

2.6 INTERACTIVE MODE & SCRIPT MODE

*In script mode an expression has no visible effect (all by itself)

- it won't display unless you tell it to

```
miles = 26.2
print miles * 1.61
```

2.7 ORDER OF OPERATIONS

- PEDMAS

2.8 STRING OPERATION

- Can't perform math operations on strings. (even if strings look li num
- Following are illegal:

```
'2'-'1'
'eggs'/'easy'
'third'*'a charm'
```

(+) works with strings to perform con-catenation
↳ joining strings by linking them end-end

```
first = 'throat'
second = 'warbler'
print first + second
```

OUTPUT

```
throatwarbler
```

(*)

```
'spam' * 3
'spamspamspam'
```

2.9 COMMENTS

- explain in natural language what the code is doing

(#) symbol comments start with
↳ can be on a line by itself or after an expressio
↳ useful when document non-obvious features

2.10 DEBUGGING

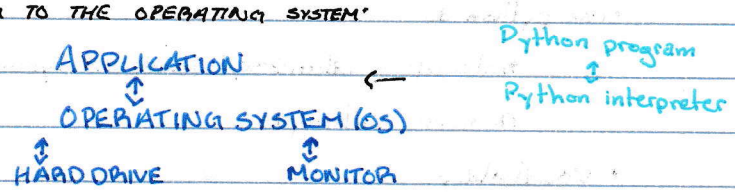
* Illegal variable names (keywords)

PRACTICAL PROGRAMMING CH2

'HELLO PYTHON'

2.1 HOW DOES A COMPUTER RUN A PYTHON PROGRAM

'TALKING TO THE OPERATING SYSTEM'



2.2 EXPRESSIONS + VALUES

(>>>) this symbol is called a prompt

// - integer division

% - modulo operator, gives us the remainder of the division.

* BE CAREFUL

using // & % w negative operands.

↳ Python takes floor of the result of an interger division

using modulo, -> sign of result matches sign of divisor.

- Operators w two operands are called binary operators
- Negation is a unary operator because applies to one operand

2.5 A SINGLE STATEMENT THAT SPANS MULTIPLE LINES

1) line breath occurs inside parentheses; OR
2) use line-continuation character, \ , backslash

SEPT 14, 2017

SYSC 1005 - LECTURE 3

```
>>> 6/2
3.0
>>> 7/2
3.5
>>> 1+2*3
7
>>> 2*3+1
7
>>> 3+7*2-5
12
>>> 5+3*3+7/2
17
```

7 or 9

* PEDMAS /symantics

* # integers