Variables and Conditionals

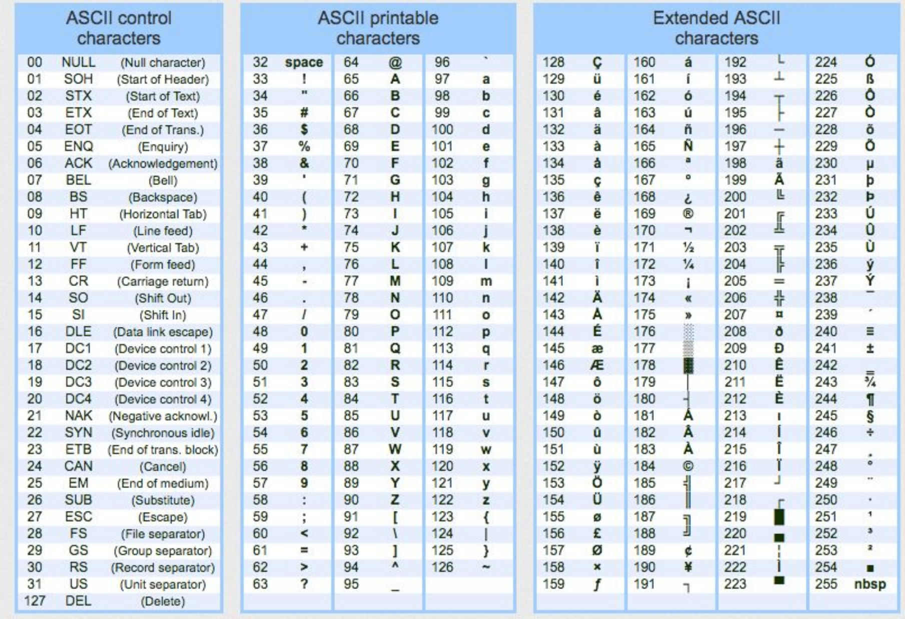
IT Adventures: Smart IT

Reading Assignment

- How to Think Like a Computer Scientist: Interactive Edition
- Required Material: 7.1 - 7.7

Basic Variables

- Integer
  - Any signed number with no decimals
  - This can make math like division look weird
  - 3/2=1 not 1.5
    - Not a problem in python 3
  - x = 5
- Float
  - Integer + decimals
  - 3/2 = 1.5
  - x = 4.2456
- Char
  - Short for character
  - Letters, symbols, and more
  - ASCII tables
  - Each character maps to an integer
  - Inside apostrophes
  - x = ‘a’



ASCII control characters			ASCII printable characters						Extended ASCII characters							
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	└	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	┴	225	ß
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	┬	226	Ô
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	├	227	Ò
04	EOT	(End of Trans.)	36	$	68	D	100	d	132	ä	164	ñ	196	─	228	õ
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	┼	229	Õ
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	å	166	ª	198	ã	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ã	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	╚	232	Þ
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	®	201	╔	233	Ú
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	╩	234	Û
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	½	203	╦	235	Ù
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	╠	236	ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ì	173	¡	205	═	237	Ý
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ä	174	«	206	╬	238	¯
15	SI	(Shift In)	47	/	79	O	111	o	143	Å	175	»	207	¤	239	´
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	░	208	ð	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	▒	209	Ð	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	▓	210	Ê	242	‗
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ô	179	│	211	Ë	243	¾
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	┤	212	È	244	¶
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	Á	213	ı	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ù	183	À	215	Î	247	¸
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	°
25	EM	(End of medium)	57	9	89	Y	121	y	153	Ö	185	╣	217	┘	249	¨
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	║	218	┌	250	·
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	╗	219	█	251	¹
28	FS	(File separator)	60	<	92	\	124	|	156	£	188	╝	220	▄	252	³
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	¢	221	¦	253	²
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	¥	222	Ì	254	■
31	US	(Unit separator)	63	?	95	_			159	ƒ	191	┐	223	▀	255	nbsp
127	DEL	(Delete)														

Strings

- A bunch of chars in row
  - Formerly an array/list of chars
  - We’ll get into arrays and lists later
  - Can still be treated like an array/list
- Lots of functions to manipulate strings, they can be found in the manual pages on the python website
- x = “foo”
  - Strings are notated by surrounding your assignment value in double quotes
- len(x)
  - Returns the length of the string x
- print(x[1])
  - Prints ‘o’
    - Arrays start at 0

Boolean

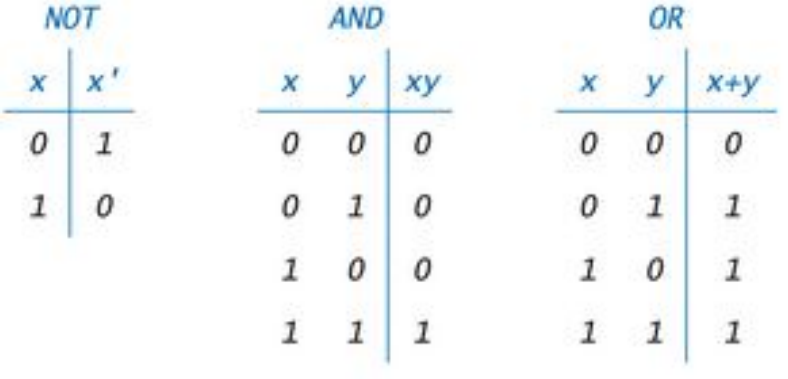
- True or False
- Any non empty or null value is True
  - Any non 0 number
  - And non empty string, list, or object
- 0 is False
  - Any number that is equal to 0
  - Any empty string, list, or object
- Works well with conditionals
- x = True
  - No apostrophes or double quotes

Conditionals

- Used to compare the values of variables
- <
    - Less than
    - 10 < 9
        - False
- ==
    - Equal to
        - = is an assignment, not a conditional
    - 10 == 9
        - False
- >
    - Greater than
    - 10 > 9
        - True

Boolean Operations

- not
  - Flips the value of the boolean expression
  - not(True) = False
- and
  - Both things being compared must be true
  - True and False = False
  - True and True = True
- or
  - Only one of the things being compared needs be true
  - False and True = True
  - False and False = False



NOT

x	x'
0	1
1	0

AND

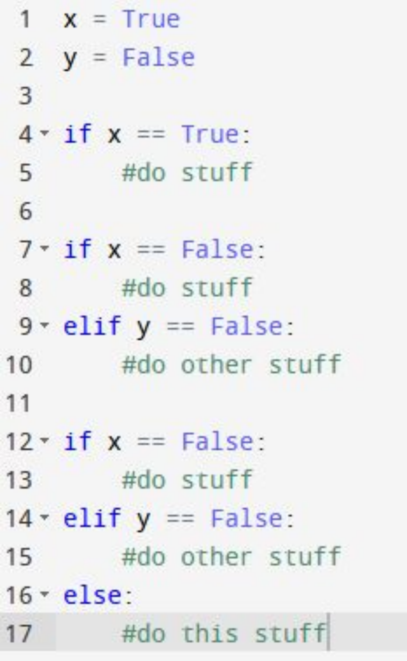
x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

If Statements

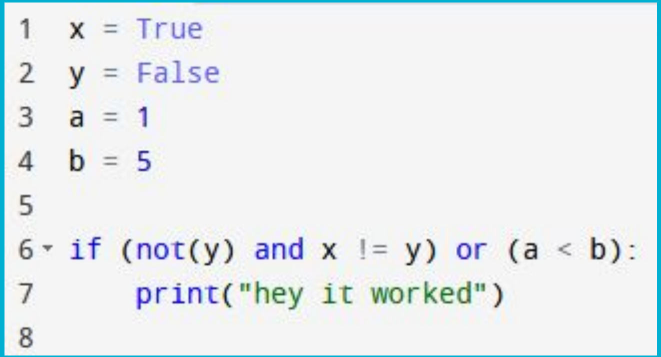
- Allows for different decisions to be made based on the value of expressions and variables
- if
    - The basic case of decision making
- elif
    - Allows for a second case to be made
    - if the if statement is false check this
- else
    - Always executes if the previous statements are false
    - not required



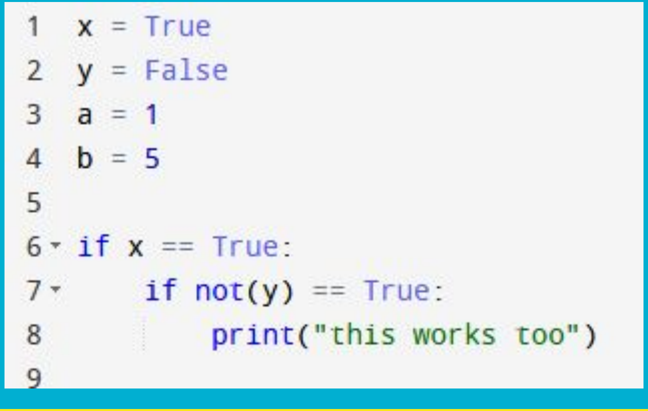
```
x = True
y = False

if x == True:
    #do stuff

if x == False:
    #do stuff
elif y == False:
    #do other stuff

if x == False:
    #do stuff
elif y == False:
    #do other stuff
else:
    #do this stuff
```

Complex if Statements

- Combine multiple conditionals in the same if statement
  - Separate expressions with parentheses to ensure you’re checking what you want to check
  - Order of operations goes left to right
- Can also be nested inside another statement



```python
x = True
y = False
a = 1
b = 5

if (not(y) and x != y) or (a < b):
    print("hey it worked")

```



```python
x = True
y = False
a = 1
b = 5

if x == True:
    if not(y) == True:
        print("this works too")

```

Technical Challenge

- Write a program that takes in an operator(+, -, *, /), and two numbers, and prints out the answer to the operation
- Hint: If not using PyCharm, you MAY need to import sys
  - We'll talk more about importing later
- If there is a typo in the operation, the code should print "Incorrect operand, try again."
- If the user enters in anything other than a number, it should be cast into an int
- Watch the YouTube link to see a step-by-step solution (after trying it first)
  - https://youtu.be/9w_Q3s49oyU