CSCI 127: Introduction to Computer Science

hunter.cuny.edu/csci

Review of Lecture 1: turtle graphics

- Imagine a turtle has a pen; when it moves forward some distance, a line is drawn on the screen.
- The turtle can also turn left some amount of degrees.

```
1   import turtle
2
3   t = turtle . Turtle ()
4
5   #draw side one
6   t . forward(100)
7   t . left (120)
8
9   #draw side two
10  t . forward(100)
11  t . left (120)
12
13  #draw side three
14  t . forward(100)
15  t . left (120)
```

Review of Lecture 1: for-loops

- The previous program used the turtle module to draw a triangle
- Rewrite the program using a for-loop

```python
1  import turtle
2
3  t = turtle.Turtle()
4
5  for i in range(3):
6      t.forward(100)
7      t.left(120)
```

For more commands, read turtle documentation

Draw a polygon with $n \geq 3$ sides

Pseudocode describes the general algorithm our program will follow; it is language-agnostic and can be translated into any programming language.

Import the turtle library

Instantiate a turtle object called t

Initialize n to be an integer ≥ 3

Repeat the following n times:

(1) t moves forward a fixed distance

(2) t turns left 360 / n degrees

Group Work: predict what will be printed

```
1
2 for j in [0,1,2,3,4,5]:
3     print(j)
4 for count in range(6):
5     print(count)
6 for color in ["red", "green", "blue"]:
7     print(color)
```

Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
  - **int**: integer or whole numbers
  - **float**: floating point or real numbers
  - **string**: sequence of characters
  - **list**: a sequence of items
    e.g. [3, 1, 4, 5, 9] or
    ["violet","purple","indigo"]
  - **class variables**: for complex objects, like turtles.
- In Python (unlike other languages) you don't need to specify the type; it is deduced by its value.

Variable Names

- There's some rules about valid names for variables.
- Can use the underscore ('_'), upper and lower case letters.
- Can also use numbers, just can't start a name with a number.
- Can't use symbols (like '+' or '*') since used for arithmetic.
- Can't use some words that Python has reserved for itself (e.g. `for`).
  (List of reserved words in *Think CS*, §2.5.)

Standardized Code for Characters

American Standard Code for Information Interchange (ASCII), 1960.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C		
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

(wiki)

ord() and chr()

(There is a link to the ASCII table on the course webpage, under "Useful Links".)

ASCII TABLE

- ord():
  input type: character
  output type: integer
- chr():
  input type: integer
  output type: character
- What is chr(33)?
- What is ord("$")?

The plus ($+$) operator for numbers and strings

- x = 3 + 5 stores the number 8 in memory location x.

- x = x + 1 increases x by 1.

- s = "hi" + "Mom" stores "hiMom" in memory locations s.

- s = s + "A" adds the letter "A" to the end of the strings s.