Remember that variables are really just pointers to objects.

```
[., cat', dog', bird', snake']
x <<<
[., cat', dog', bird', snake']
head = [2] x <<<
[., cat', dog', bird', snake']
x <<<
[., cat', dog', bird', snake']
head <<<
```

Y points to a new list

and X points to the same object.

When you need to make a copy of a list so that the old list and the new list are separate, you should use [:] which is a shorthand for taking a slice from 0 to length. Slices are new copies.

A hint

```
W: L X
\[ \quad 1 + 2 \]
M = Z
\[ \quad 2 - 3 \]
```

Why functions should be short.

I may also ask you anything in the notes that I delivered in previous classes. So always review those.

Remember that anything in the Zybooks is eligible for the Buffalo nickel question. Stop quiz.

---

<table>
<thead>
<tr>
<th>Finish Stretching Chs. 10, 11 and 13</th>
<th>Textbook-assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 10 and 11</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Python</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
How to create your own exception class
How to throw an exception
Basic ideas
See the seli-quizzter and the Z3books

C. Exceptions

I could only import a package module by module.
If you import the package, all the modules in it are imported (supposedly).
A package is a directory with modules in it. plan a file

interact with the system to get some initial values
create some objects
set up some variables
read some values from a file

imported, such code might:

But if you need initialization code, don't "import" this way. Let it run when the module is

This code is often misleading code, i.e. code that executes the functions in this script

if __name__ == '__main__':
    ...

Done:

If there is code in the module that you don't want run when it is imported, then project it by

Modules can be run as standalone scripts or imported.

B. More about modules
2. Runtime error - you print "1st step

3. Logical error

4. System error

Hacking error

30 crashes

\[ x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \]
Recurrence relation, e.g., Fibonacci sequence: $F_n = F_{n-1} + F_{n-2}$

**Recursion**

**Examples:**

* What is recursion? How is it used? Why use it? (1 point)

There are two basic functions in Python:

* `exec` command

```python
exec(command)
```

* `eval` expression

```python
eval(command)
```

There are 2 primitive functions in Python:

* Chop off the leading #

```python
exec('command'[:])
```

```python
eval('command'.startswith('""'))
```

Very dangerous, play this code before putting it into production.

Create a backdoor for debugging, *CMD*.

Use a try/except to catch errors so it won't crash the program.

D. Extra ideas for a command-line project