

2.2.1 – Data type of var1?

It is a data type within the class of integer, so it is a int

```
>>> type(var1)
<class 'int'>
>>>
```

2.2.2 – Interpreters knows data type, what does it indicate about type binding that python uses?

Python must be using a dynamic type binding, since it determines it's data type at run time.

2.3.1 – What is the value assigned to variable var2?

It is 28.25

```
>>> print(var2)
28.25
>>>
```

2.3.2 What's the data type of var 2?

It is a data type within the class of float, therefore it's a float.

```
>>> type(var2)
<class 'float'>
>>>
```

2.3.3 var1 and var2 have different data types. What's the characteristics of data type conversions in Python does this indicate?

This indicates the use of conversion was done, or that specifically implicit conversion was done, since var2 is a float multiplied by an int.

2.4.1 what data type is str1?

Str1 is a datatype in the class of string, therefore it's a string

```
>>> type(str1)
<class 'str'>
>>>
```

2.5.1 What is the output of str1 + var1 + var2?

An error. Strings can only concatenate with strings, and will not convert to int accordingly.

It is a type error.

```
>>> str1 + var1 + var2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
>>>
```

2.6.1 What is the output of `int(str1) + var1 + var2`

It is 43.25.

```
>>> int(str1) + var1 + var2
43.25
>>> 
```

There was no error, since all of the data types were the same, thus no type error occurred.

2.7.1 What is `str1 + str(var1) + str(var2)`?

It is '10528.25'

```
>>> str1 + str(var1) + str(var2)
'10528.25'
>>> 
```

2.7.2 What operation does this expression seem to be doing?

It is concatenating the strings, or simply sticking the strings together.