Variables and Data Types

GITAA
Transforming careers

In this lecture

- Naming variables
- Basic data types
  - Identify data type of an object
  - Verify if an object is of a certain data type
  - Coerce object to new data type

Naming variables

- Values assigned to variables using an assignment operator '='
- Variable name should be short and descriptive
  - Avoid using variable names that clash with inbuilt functions
- Designed to indicate the intent of its use to the end user
- Avoid one character variable names
  - One character variable names are usually used in looping constructs, functions, etc

Naming variables

- Variables can be named alphanumerically

```
Age=55   age=55   age2=55   Age2=55
```

- However the first letter must start with an alphabet (lowercase or uppercase)

```
In [3]: 2age=55
  File "<ipython-input-3-00efac86f1ae>", line 1
    2age=55
       ^
SyntaxError: invalid syntax
```

Naming variables

- Other special character
  - Underscore (_)

```
Employee_id=501
```

  - Use of any other special character will throw an error

```
In [6]: Employee@id=501
  File "<ipython-input-6-e2ec8cb31ebe>", line 1
    Employee@id=501
                ^
SyntaxError: can't assign to operator
```

  - Variable names should not begin or end with underscore even though both are allowed

```
_age=55
age_=55
```

Naming conventions

- Commonly accepted case types
  - Camel (lower and upper)

```
ageEmp=45        AgeEmp=45
```

  - Snake

```
age_emp=45       Age_emp=45
```

  - Pascal

```
AgeEmp=45
```

Assigning values to multiple variables

Code

```
Physics,Chemistry,Mathematics=89,90,75
```

Values reflected in environment

Variable explorer

Name	Type	Size	Value
Chemistry	int	1	90
Mathematics	int	1	75
Physics	int	1	89

Data types

Basic data types

Basic data types	Description	Values	Representation
Boolean	represents two values of logic and associated with conditional statements	True and False	bool
Integer	positive and negative whole numbers	set of all integers, Z	int
Complex	contains real and imaginary part (a+ib)	set of complex numbers	complex
Float	real numbers	floating point numbers	float
String	all strings or characters enclosed between single or double quotes	sequence of characters	str

Identifying object data type

- Find data type of object using
- Syntax: `type(object)`

```
Employee_name="Ram"
Age=55
Height=150.6
```

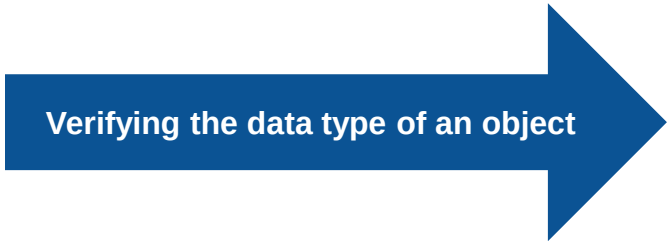
Checking the data type of an object

```
In [10]: type(Employee_name)
Out[10]: str

In [11]: type(Age)
Out[11]: int

In [12]: type(Height)
Out[12]: float
```

Verifying object data type

- Verify if an object is of a certain data type
- Syntax: `type(object) is datatype`

```
Employee_name="Ram"
Age=55
Height=150.6
```




```
In [13]: type(Height) is int
Out[13]: False

In [14]: type(Age) is float
Out[14]: False

In [15]: type(Employee_name) is str
Out[15]: True
```

Coercing object to new data type

- Convert the data type of an object to another
- Syntax: `datatype(object)`
- Changes can be stored in same variable or in different variable

```
Employee_name="Ram"
Age=55
Height=150.6
```

Coercing the data type of an object

```
In [16]: type(Height)
Out[16]: float

In [17]: ht=int(Height)

In [18]: type(ht)
Out[18]: int

In [19]: Height=int(Height)

In [20]: type(Height)
Out[20]: int
```

Coercing object to new data type

- Only few coercions are accepted
- Consider the variable '*Salary_tier*' which is of string data type
- '*Salary_tier*' contains an integer enclosed between single quotes

```
Salary_tier='1'
```

Coercing the data type of an object

```
In [20]: type(Salary_tier)
Out[20]: str

In [21]: Salary_tier=int(Salary_tier)

In [24]: type(Salary_tier)
Out[24]: int
```

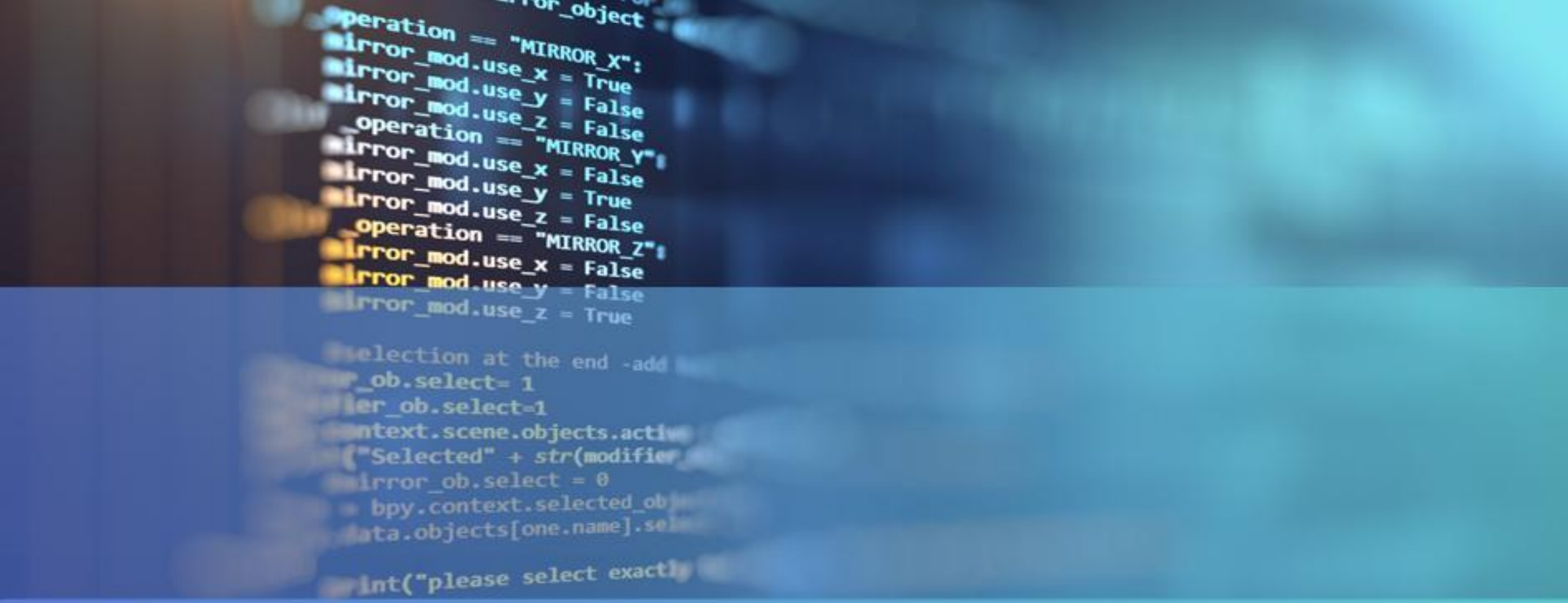
Coercing object to new data type

- However if the value enclosed within the quotes is a string then conversions will not be possible

```
In [19]: Employee_name="Ram"

In [20]: Employee_name=float(Employee_name)
Traceback (most recent call last):

  File "<ipython-input-20-b0286eb77de0>", line 1, in <module>
    Employee_name=float(Employee_name)

ValueError: could not convert string to float: 'Ram'
```

Summary

- Conventions to name a variable
- Basic data types
  - Get data type of a variable
  - Verify if a variable is of a certain data type
  - Coerce variable to new data type


operation == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True
selection at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.acti
("Selected" + str(modifier
irror_ob.select = 0
bpy.context.selected_ob
ata.objects[one.name].se
int("please select exact

THANK YOU