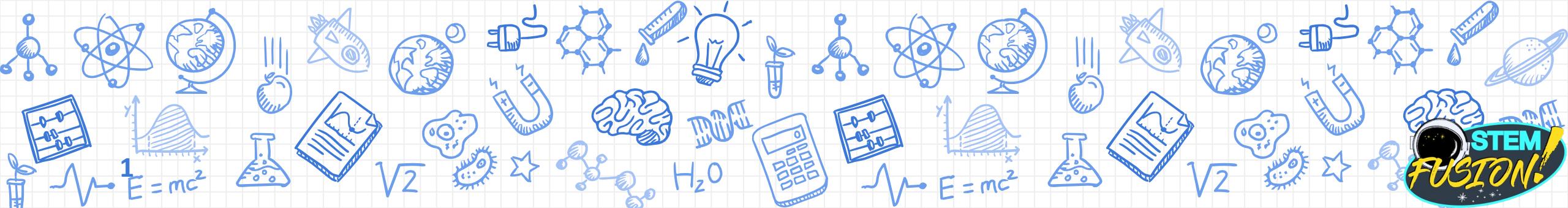


RESTART YOUR COMPUTER



Introduction to Computational Thinking and Python Programming

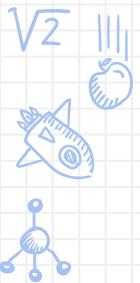
Sarom Leang, Ph.D. (Instructor)

Jesse McCandlish (Mentor)

April 22, 2023

Session 5





[REMINDER]

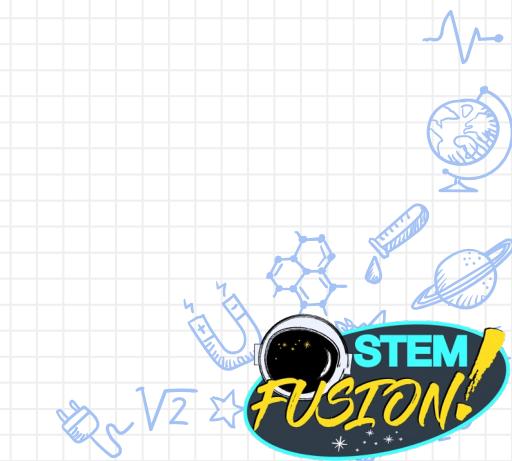
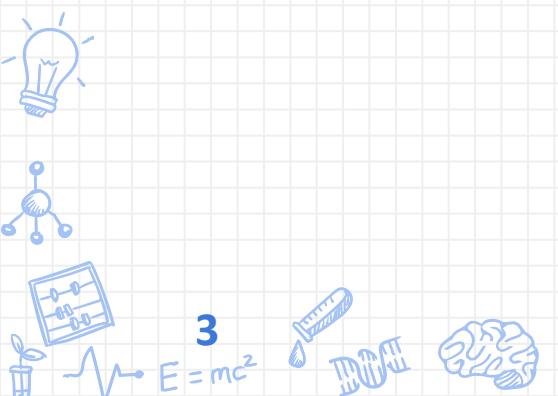
Names and Faces and Pronouns

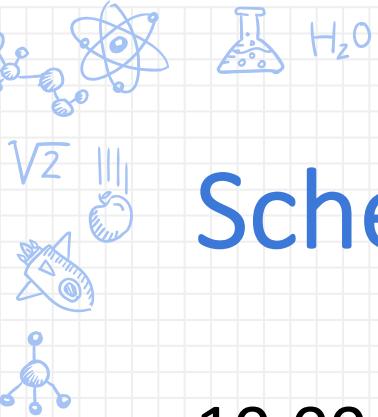
Sarom Leang, Ph.D. (Instructor)

- Professor
- Instructor
- Mr. Leang

Jesse McCandlish (Mentor)

- Jesse





Schedule

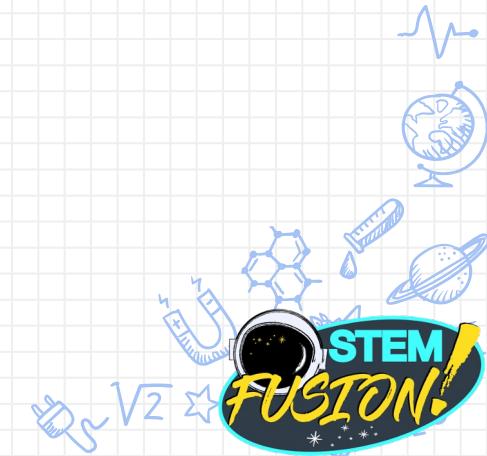
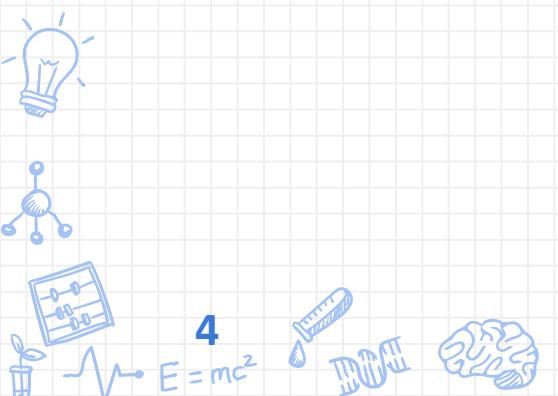
10:00 AM – 10:15 AM Homeroom

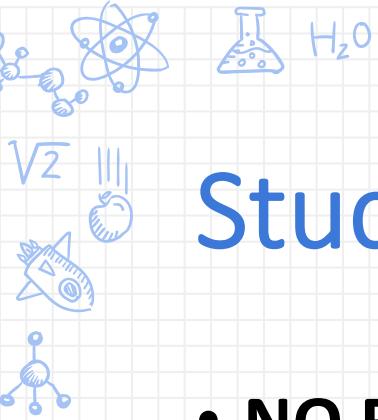
10:15 AM – 11:35 AM G1 Block

11:35 AM – 12:35 PM Break/Lunch

12:40 PM – 02:00 PM G2 Block

80 minutes of class time

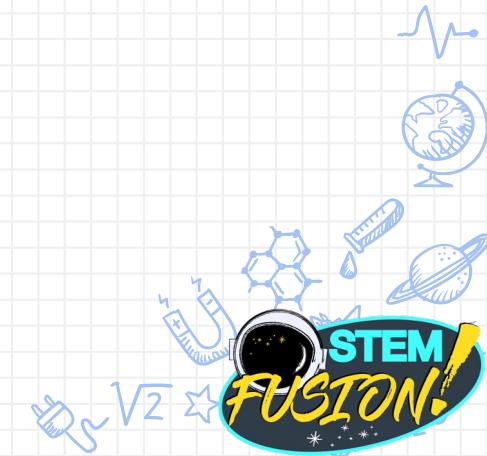


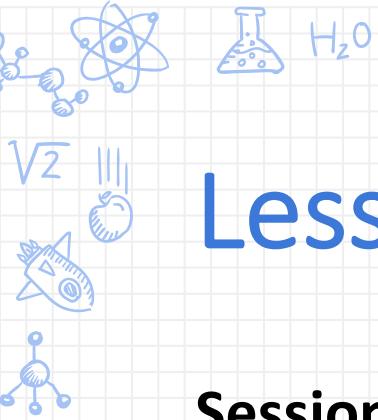


[REMINDER]

Student Expectations

- **NO FOOD**
- **NO DRINKS** (on the table)
- Be respectful to individuals and property
- Be open to learning
- Be open to not understanding
- Be patient with yourself
- Ask questions
- Explore
- **Embrace failure**





[UPDATED]

Lesson Plan

Session #1 – October 29, 2022

- Entrance survey

Session #2 – December 3, 2022

- Parallel computing
- Python programming environment, “Hello World”

Session #3 – March 4, 2023

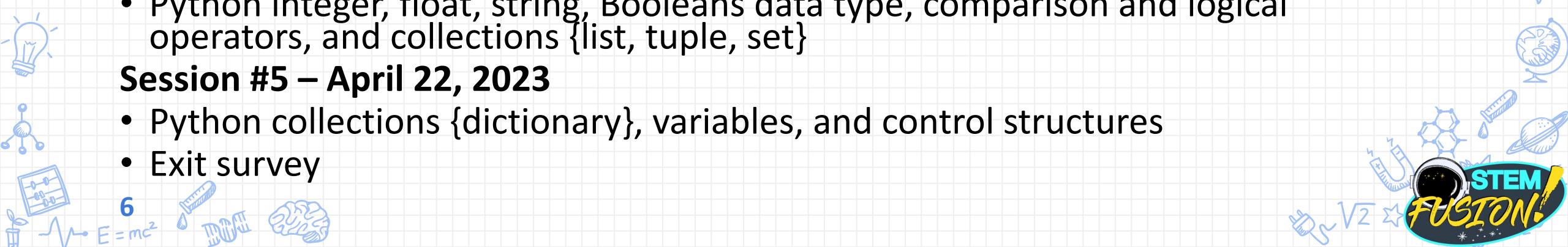
- Generative AI
- Python arithmetic operators

Session #4 – March 25, 2023

- Python integer, float, string, Booleans data type, comparison and logical operators, and collections {list, tuple, set}

Session #5 – April 22, 2023

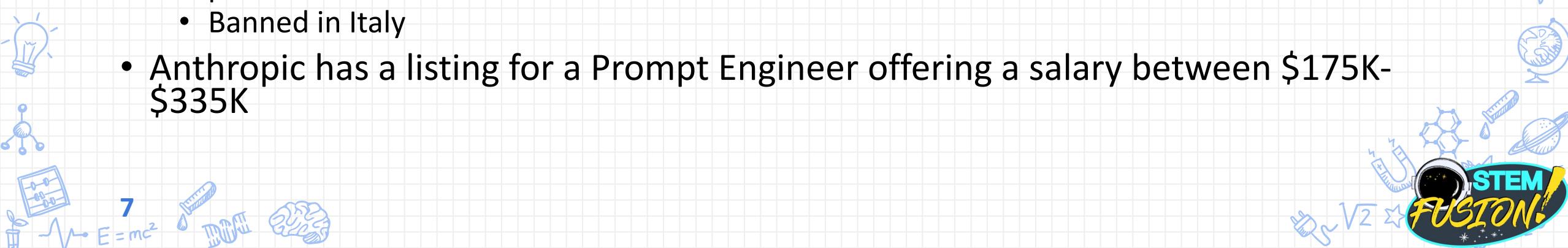
- Python collections {dictionary}, variables, and control structures
- Exit survey





Technology in the News (Generative AI)

- Concerns of economic and social impact of generative AI.
- Buzzfeed News closes down.
- Elon Musk and Apple cofounder Steve Wozniak among over 1,100 who signed open letter calling for 6-month ban on creating powerful AI.
- Copyright concerns (DALL-E, artificial vocals):
 - Can copyright subsist in AI-generated content?
 - Does the use of generative AI models infringe the copyright in pre-existing works?
- Privacy concerns:
 - Compliance with General Data Protection Regulation (GDPR) on use, process, and storage of personal data
 - Banned in Italy
- Anthropic has a listing for a Prompt Engineer offering a salary between \$175K-\$335K



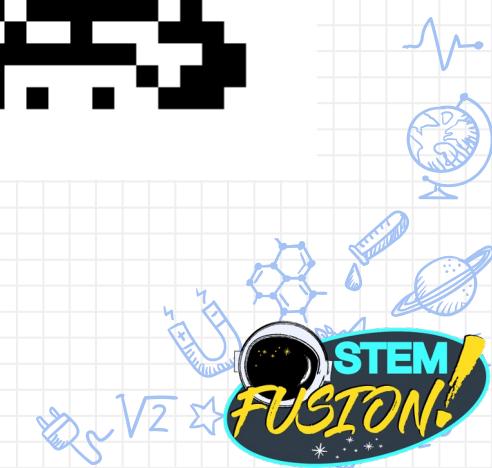
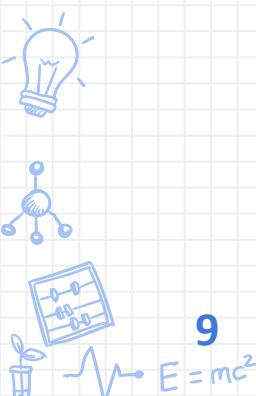
Generative AI and Academic Code of Conduct

- Cheating or violating academic code of conduct using generative AI is unethical and has serious consequences.
- Disciplinary action may include failing the assignment or course, or even expulsion from school.
- Long-term consequences may include damaging a student's reputation and future academic and career prospects.
- **Use generative AI responsibly and in accordance with the academic code of conduct.**



Generative AI DALL-E Competition

- Visit: <https://bit.ly/EIP23-G1>
- Password: python
- Click on the image you like the most
- Enter your name
- Click Proceed as Guest



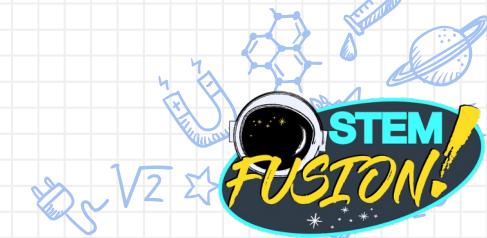
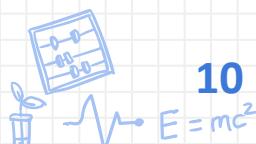


Generative AI DALL-E Competition

- Visit: <https://bit.ly/EIP23-G2>
- Password: python
- Click on the image you like the most
- Enter your name
- Click Proceed as Guest



$$E = mc^2$$



STEM
FUSION!

Basic Data Types in Python

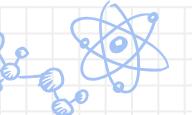
- **Integer:** No decimal, `0, -1, 1, -100, 100,`
- **Floating-point:** With a decimal, `0.0, -1.0, 1.0, -100.0, 100.0`
- **String:** An array of characters between `'...', "..."`, or `'''...'''`
- **Boolean:** `True, False`
- **List:** ordered, mutable, contained in `[..., ...]`
- **Array:** ordered, mutable, contained in `[..., ...]`, same data type
- **Tuple:** ordered, immutable, contained in `(..., ...)`
- Set
- Dictionary



$$\sqrt{2}$$

$$V_2$$

$$H_2O$$



H₂O

$\sqrt{2}$



Set

- Mutable (changeable)
- Unordered collection
- Each element is unique
- Contained within curly braces { }
- Can not be indexed **[-1]** or sliced **[:-1]**
- Can perform set operations:
 - Union
 - Intersection
 - Difference



12



$$E=mc^2$$





$$\sqrt{2}$$



Set

```
s1 = {'apple', 'banana', 'carrot', 'daikon'}
```

```
s2 = {'daikon', 'eggplant', 'fig', 'grapes'}
```

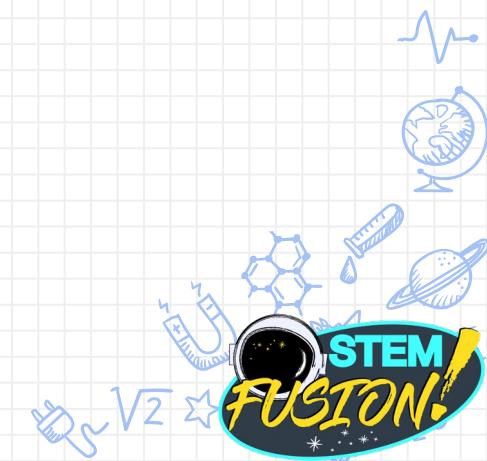
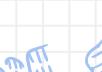
```
s1.union(s2)
```

```
s1.intersection(s2)
```

```
s1.difference(s2)
```



13





Set

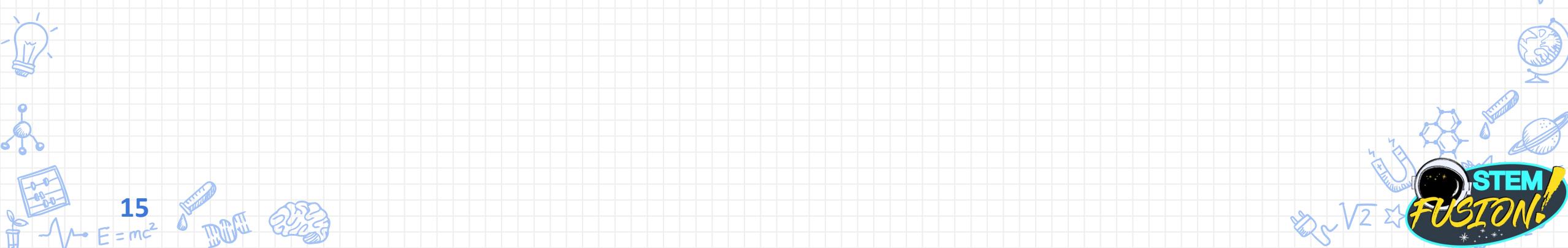
- Set Theory Notation (set theory is an area of mathematics)
 - Union: $A \cup B$
 - Intersection: $A \cap B$
- Many more operations
 - Symmetric difference: $A \ominus B$: `symmetric_difference`
 - Disjoint: $A \cup B = \emptyset$: `isdisjoint`
 - Subset: $A \subseteq B$: `issubset`
 - Superset: $A \supseteq B$: `issuperset`

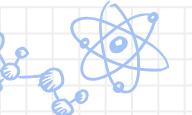




Dictionary

- Mutable (changeable)
- Unordered collection of **key-value** pairs
 - "color" : "red"
 - "is_heavy" : True
 - "cost" : 4.99
- Each **key** is unique
- Contained within curly braces { }





$$\sqrt{2}$$



Accessing a Value in a Dictionary

```
{"color": "red", "is heavy": True, "cost": 4.99}["color"]  
{"color": "red", "is heavy": True, "cost": 4.99}["is heavy"]  
{"color": "red", "is heavy": True, "cost": 4.99}["cost"]  
{"color": "red", "is heavy": True, "cost": 4.99}[0]
```

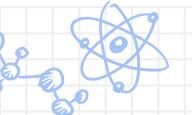
```
{"colors": ["red", "white", "blue"]}["colors"]  
{"colors": ["red", "white", "blue"]}["colors"][0]
```

```
[{"name": "Alice"}, {"name": "Bob"}][0]  
[{"name": "Alice"}, {"name": "Bob"}][0]["name"]
```



16





H₂O

$\sqrt{2}$



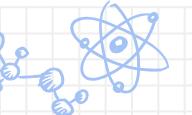
Dictionary Within a Dictionary

```
{  
    "state": {  
        "AL": "Alabama",  
        "AK": "Alaska",  
        "AZ": "Arizona" },  
    "territory": {  
        "AS": "American Samoa",  
        "GU": "Guam",  
        "VI": "Virgin Islands" }  
}
```



$$17 \quad E=mc^2$$





H₂O

$\sqrt{2}$



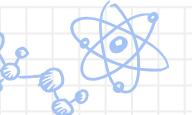
Dictionary Within a Dictionary

```
{  
    "state": {  
        "AL": "Alabama",  
        "AK": "Alaska",  
        "AZ": "Arizona" },  
    "territory": {  
        "AS": "American Samoa",  
        "GU": "Guam",  
        "VI": "Virgin Islands" }  
}["territory"]
```



18





H₂O

$\sqrt{2}$

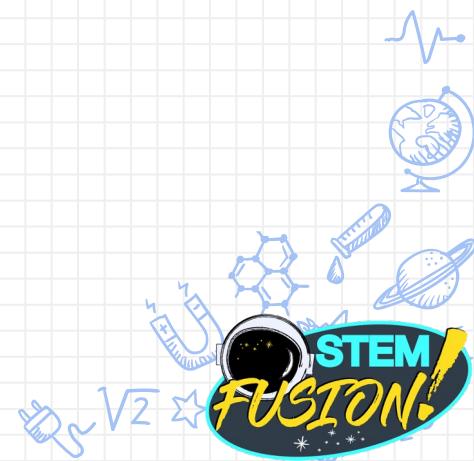


Dictionary Within a Dictionary

```
{  
    "state": {  
        "AL": "Alabama",  
        "AK": "Alaska",  
        "AZ": "Arizona" },  
    "territory": {  
        "AS": "American Samoa",  
        "GU": "Guam",  
        "VI": "Virgin Islands" }  
}["territory"]["GU"]
```



19

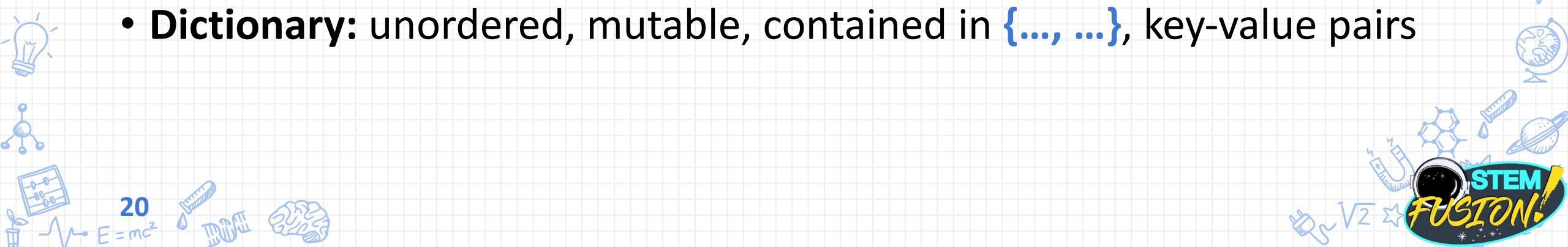


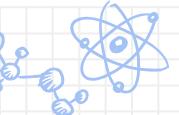
STEM
FUSION!



Basic Data Types in Python

- **Integer:** No decimal, **0, -1, 1, -100, 100,**
- **Floating-point:** With a decimal, **0.0, -1.0, 1.0, -100.0, 100.0**
- **String:** An array of characters between '**...', "...**', or **'''...'''**
- **Boolean:** **True, False**
- **List:** ordered, mutable, contained in **[..., ...]**
- **Array:** ordered, mutable, contained in **[..., ...]**, same data type
- **Tuple:** ordered, immutable, contained in **(..., ...)**
- **Set:** unordered, mutable, contained in **{..., ...}**, elements are unique
- **Dictionary:** unordered, mutable, contained in **{..., ...}**, key-value pairs





H₂O

$\sqrt{2}$



?

{

"state": {

 "AL": "Alabama",

 "AK": "Alaska",

 "AZ": "Arizona" },

"territory": {

 "AS": "American Samoa",

 "GU": "Guam",

 "VI": "Virgin Islands" }

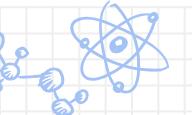
}["state"]["CA"] = "California"



21

$$E=mc^2$$





H₂O

$\sqrt{2}$



Variables

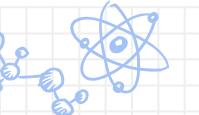


- Similar to variables in math
- Used to store values (e.g., integer, float, string, Boolean, list, etc.)
- Can be used later in a program to refer to its **last assigned** value
- You can give a variable almost any name you like
 - Case sensitive (`age`, `Age`, and `AGE` are three different variables)
 - Can only contain letters (`A-Z`, `a-z`), numbers (`0-9`), and underscores (`_`)
 - Can NOT begin with a number
 - Helpful if variable names are descriptive (`a` vs. `age`)



$$22 \quad E = mc^2$$





H₂O

$\sqrt{2}$



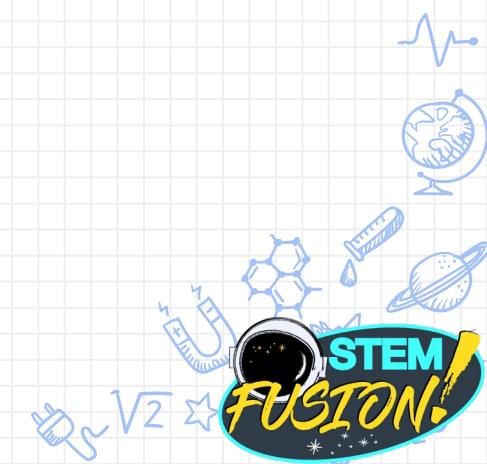
Variables



Valid	Invalid
age	1age
age	age*
age_	+age
_0age	@ge



23





Variable Conventions

- Camel Case
 - Capitalized all words following the first word and removing spaces
 - date of birth in Camel Case is `dateOfBirth`
- Pascal Case
 - Capitalized all words and removing spaces
 - date of birth in Pascal Case is `DateOfBirth`
- Snake Case
 - Replace each space with an underscore using either all upper- or lowercase
 - date of birth in Snake Case is `date_of_birth`

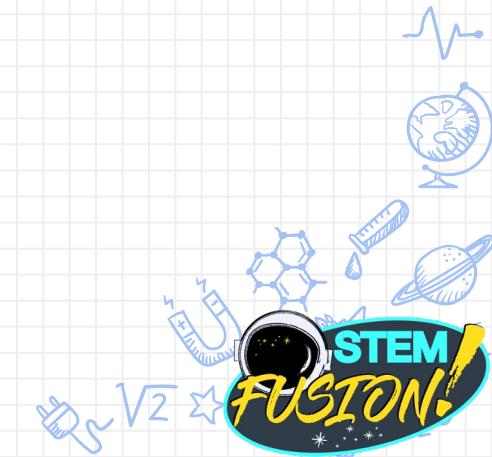


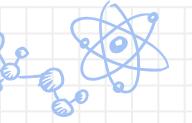


Variable Assignment

- Use the assignment operator, `=`, to storage information in a variable
- Variable being assigned a value is on the left-hand side of `=`
- Value being assigned to the variable is on the right-hand side of `=`

```
date_of_birth = "04/22/2023"
```





H₂O

$\sqrt{2}$



?

```
dictionary={  
    "state": {  
        "AL": "Alabama",  
        "AK": "Alaska",  
        "AZ": "Arizona" },  
    "territory": {  
        "AS": "American Samoa",  
        "GU": "Guam",  
        "VI": "Virgin Islands" }  
}  
dictionary["state"]["CA"] = "California"  
print(dictionary)
```



26





Hello World! -> Hello <name>!

- Modify the "Hello World" program to print out "Hello <name>" where <name> is the value stored in the variable `name`

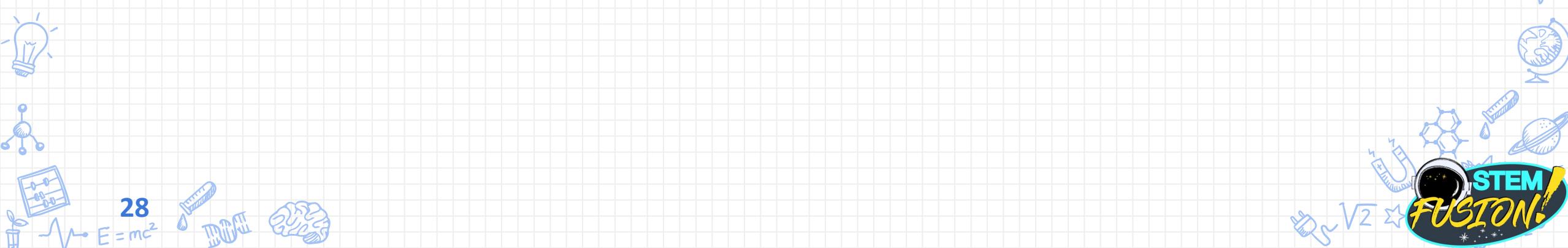


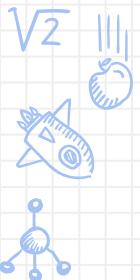


Hello World! -> Hello <name>!

- Modify the "Hello World" program to print out "Hello <name>" where <name> is the value stored in the variable `name`
- Solution (`hello-name.py`):

```
name = "Hal"  
print("Hello", name, "!")
```

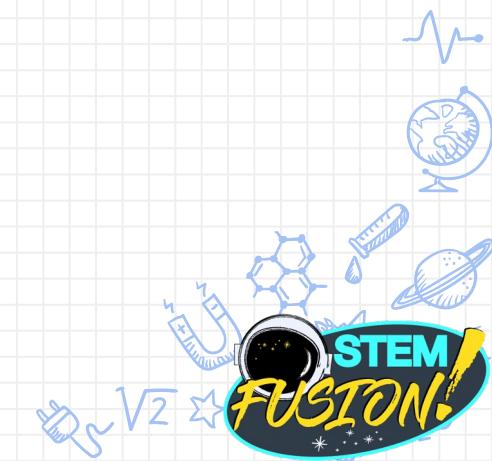
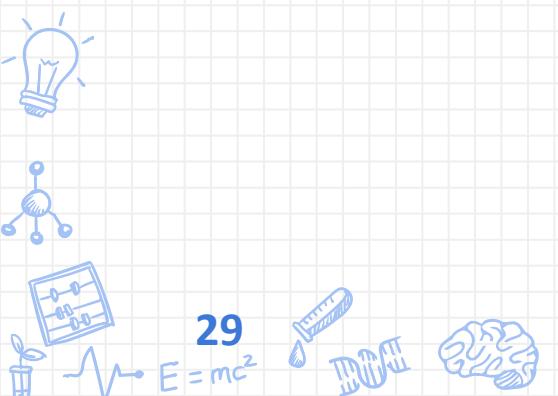




Hello World! -> Hello <name>!

- Allow users to provide the input for the variable name.
- Solution (hello-you.py):

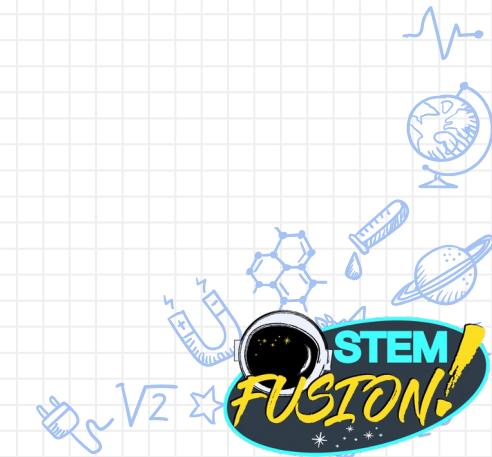
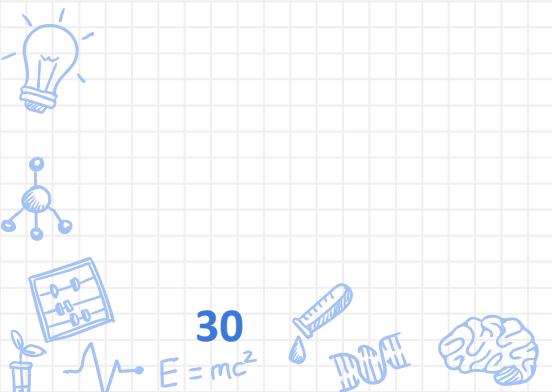
```
name = input("What is your name? ")  
print("Hello", name, "!" )
```





Control Flow and Control Structures

- Control flow is the order in which a program's code executes.
- Regulated by **conditional statements, loops, and function calls**.
- In the absence of control structures, Python code is executed in order from top to bottom.

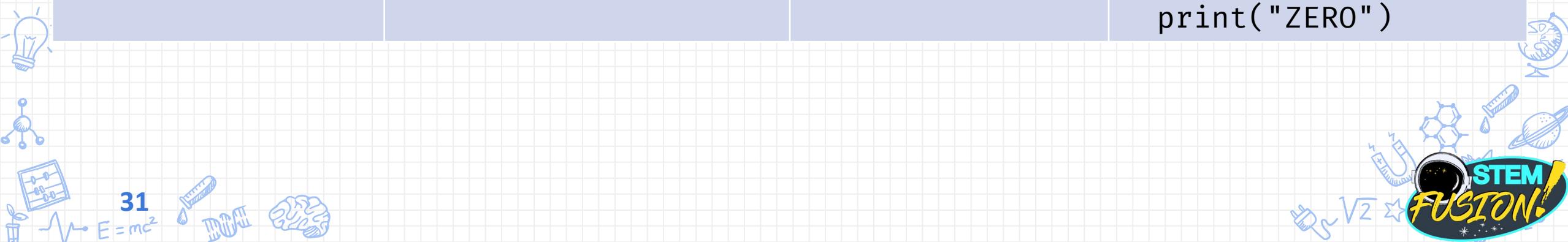


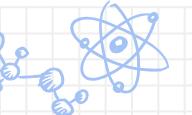


Conditional Statements

- Execute certain code blocks based on whether a specific condition is **True** or **False**

if	Nested if	if-else	if-elif-else
<pre>n = 10 if n % 2 == 0: print("even")</pre>	<pre>n = -10 if n % 2 == 0: if n < 0: print("- even") if n > 0: print("+ even")</pre>	<pre>n = 10 if n % 2 == 0: print("even") else: print("odd")</pre>	<pre>n = 0 if n > 0: print("positive") elif n < 0: print("negative") else: print("ZERO")</pre>





H₂O

$\sqrt{2}$



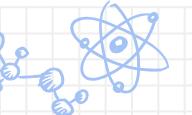
Selective Hello <name>!

- Only say Hello for a specific value of name.



32





H₂O

$\sqrt{2}$



Selective Hello <name>!

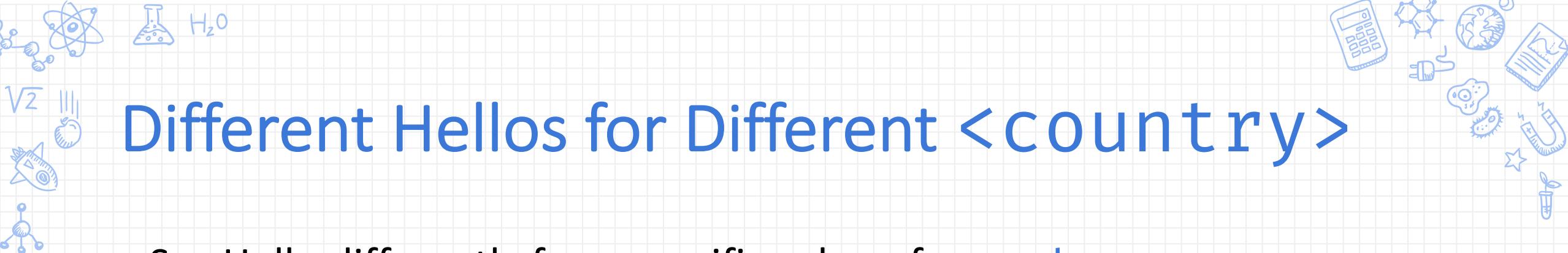
- Only say Hello for a specific value of name.
- Solution (hello-who-dis.py):

```
name = input("What is your name? ")  
if name == "Hal":  
    print("Hello", name, "!")  
else:  
    print("New phone, who dis?")
```



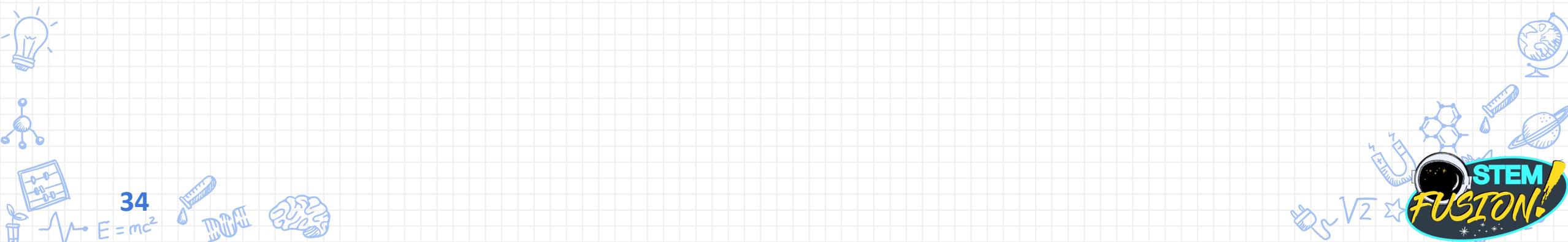
33

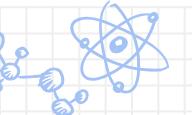




Different Hellos for Different <country>

- Say Hello differently for a specific value of **country**.





H₂O

$\sqrt{2}$



Different Hellos for Different <country>

- Say Hello differently for a specific value of **country**.
- Solution (hello-country):

```
country = input("What country? ")  
if country == "USA":  
    print("Hello world!")  
elif country == "Mexico":  
    print("Hola Mundo")  
elif country == "Japan":  
    print("Kon'nichiwa sekai!")  
else:  
    print("Greetings Earthling!")
```



$$E=mc^2$$

35





H₂O

$\sqrt{2}$

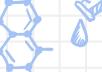
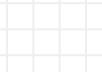
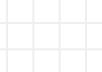
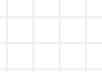
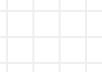
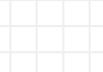
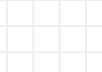
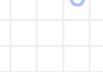
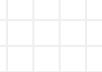


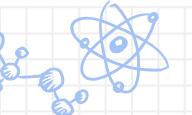
?

```
country = input("What country? ")
if country == "USA":
    print("Hello world!")
elif country == "Mexico":
    print("Hola Mundo")
elif country == "Japan":
    print("Kon'nichiwa sekai!")
elif country == "Japan":
    print("「こんにちは世界」")
else:
    print("Greetings Earthling!")
```



36





H₂O

$\sqrt{2}$



Repetition Statements

- Used to repeat a block of instructions (for, while)

for	while
<pre>for i in range(0,10): print(i)</pre>	<pre>j = 10 i = 0 while (i < 10): print(i) i=i+1</pre>



37





Iterate Over Collections

list	tuple	set
<pre>x=[1,2,3,4,5] for i in range(len(x)): print(x[i])</pre>	<pre>x=(1,2,3,4,5) for i in range(len(x)): print(x[i])</pre>	<pre>x={1,2,3,4,5} for i in range(len(x)): print(x[i])</pre>

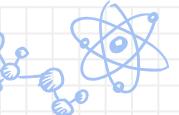




Iterate Over Collections

list	tuple	set
<pre>x=[1,2,3,4,5] for i in range(len(x)): print(x[i])</pre>	<pre>x=(1,2,3,4,5) for i in range(len(x)): print(x[i])</pre>	<pre>x={1,2,3,4,5} for i in range(len(x)): print(x[i]) for i in x: print(x)</pre>





H₂O

$\sqrt{2}$



Goodbye Everyone!

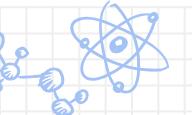
- Iterate over a collection of everyone's name and say Goodbye!



40

$$E = mc^2$$





H₂O

✓²



Goodbye Everyone!

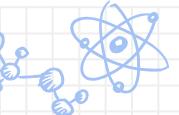
- Iterate over a collection of everyone's name and say Goodbye!
- Solution (goodbye-everyone.py)

```
names=[...]
for i in range(len(names)):
    print("Goodbye", names[i], "!" )
```



41





H₂O

$\sqrt{2}$



Exit Survey

- G1 – bit.ly/Exit23-G1
- G2 – bit.ly/Exit23-G2



42

$$E=mc^2$$

100

100

