Presentation 3

Functions & Modules

Announcements

Reminders

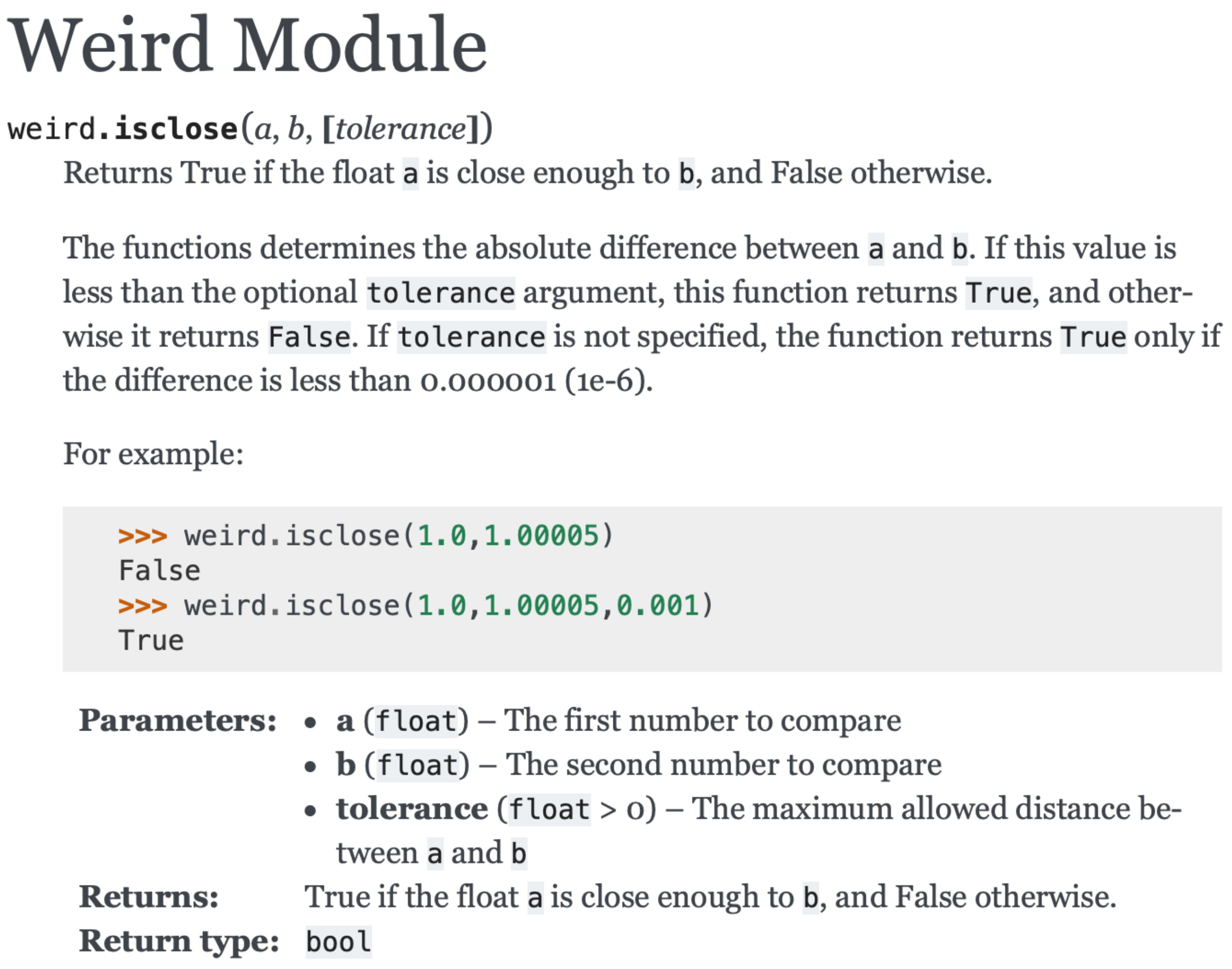
- Have graded **AI quiz**
  - Take now if have not
  - If made 9/10, are okay
  - Else must retake
- **Survey 0** is still open
  - For participation score
  - Must complete them
- Must access in CMS

Keeping Up With Videos

- **Today**
  - **Lesson 3:** Function Calls
  - **Lesson 4:** Modules
  - Videos 4.1-4.5
- Next Time
  - Video 4.6 of **Lesson 4**
  - **Lesson 5:** Function Defs
- Also *skim* Python API

Questions?

Reading Documentation



Weird Module

weird.**isclose**(*a*, *b*, [*tolerance*])

Returns True if the float `a` is close enough to `b`, and False otherwise.

The functions determines the absolute difference between `a` and `b`. If this value is less than the optional `tolerance` argument, this function returns `True`, and otherwise it returns `False`. If `tolerance` is not specified, the function returns `True` only if the difference is less than 0.000001 (1e-6).

For example:

```
>>> weird.isclose(1.0,1.00005)
False
>>> weird.isclose(1.0,1.00005,0.001)
True
```

Parameters:
- **a** (`float`) – The first number to compare
- **b** (`float`) – The second number to compare
- **tolerance** (`float` > 0) – The maximum allowed distance between `a` and `b`

Returns: True if the float `a` is close enough to `b`, and False otherwise.

Return type: `bool`

Reading isclose

- Assume that we type

```
>>> import weird
>>> isclose(2.000005,2.0)
```

- What is the result (value)?

A: `True`
B: `False`
C: An error!
D: Nothing!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> isclose(2.000005,2.0)
```

- What is the result (value)?

A: `True`
B: `False`
C: An error! **CORRECT**
D: Nothing!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.isclose(2.000005,2.0)
```

- What is the result (value)?

A: `True`
B: `False`
C: An error!
D: Nothing!
E: I do not know

Reading `isclose`

- Assume that we type

```
>>> import weird
>>> weird.isclose(2.000005,2.0)
```

- What is the result (value)?

A: `True`
B: `False` **CORRECT**
C: An error!
D: Nothing!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.isclose(2.0,3.0,4.0)
```

- What is the result (value)?

A: `True`
B: `False`
C: An error!
D: Nothing!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.isclose(2.0,3.0,4.0)
```

- What is the result (value)?

A: `True` **CORRECT**
B: `False`
C: An error!
D: Nothing!
E: I do not know

`weird.skipto`(*value*, [*period*, [*start*]])

Returns the nearest integer after `value` within the specified sequence.

By default, this function returns the nearest non-negative even number after `value`. For example:

```
>>> weird.skipto(0.5)
2
>>> weird.skipto(-3)
0
```

If the optional argument `period` is specified, it will use the next non-negative multiple of `period` instead of an even number. That is, for a period of *p*, the sequence will be *0*, *p*, *2*p* and so on.

```
>>> weird.skipto(0.5,3)
3
>>> weird.skipto(6.5,5)
10
```

The optional argument `start` indicates the place to start counting when you add multiples of `period`. By default is it 0 (hence the limitation of nonnegative values). For example, if `period` is 2 and `start` is 1, you would use odd values instead of even values. Here are some more explicit examples:

```
>>> weird.skipto(2,3,1)
4
>>> weird.skipto(-3,2,-2)
-2
```

Parameters:
- **value** (`int` or `float`) – The value to skip from
- **period** (`int`) – The multiple of the skip sequence
- **start** (`int`) – The start of the skip sequence

Returns: The nearest integer after `value` within the specified sequence.

Return type: `int`

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.skipto(3.5)
```

- What is the result (value)?

A: 2
B: 3
C: 4
D: An error!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.skipto(3.5)
```

- What is the result (value)?

A: 2
B: 3
C: 4 **CORRECT**
D: An error!
E: I do not know

Reading skipto

- Assume that we type

```
>>> from weird import *
>>> skipto(3.5)
```

- What is the result (value)?

A: 2
B: 3
C: 4
D: An error!
E: I do not know

Reading isclose

- Assume that we type

```
>>> from weird import *
>>> skipto(3.5)
```

- What is the result (value)?

A: 2
B: 3
C: 4 **CORRECT**
D: An error!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.skipto(3.5,6)
```

- What is the result (value)?

A: 6
B: 5
C: 4
D: An error!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.skipto(3.5,6)
```

- What is the result (value)?

A: 6 **CORRECT**
B: 5
C: 4
D: An error!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.skipto(3.5,2,1)
```

- What is the result (value)?

A: 6
B: 5
C: 4
D: An error!
E: I do not know

Reading isclose

- Assume that we type

```
>>> import weird
>>> weird.skipto(3.5,2,1)
```

- What is the result (value)?

A: 6

B: 5 **CORRECT**

C: 4

D: An error!

E: I do not know