DISCUSSION 01

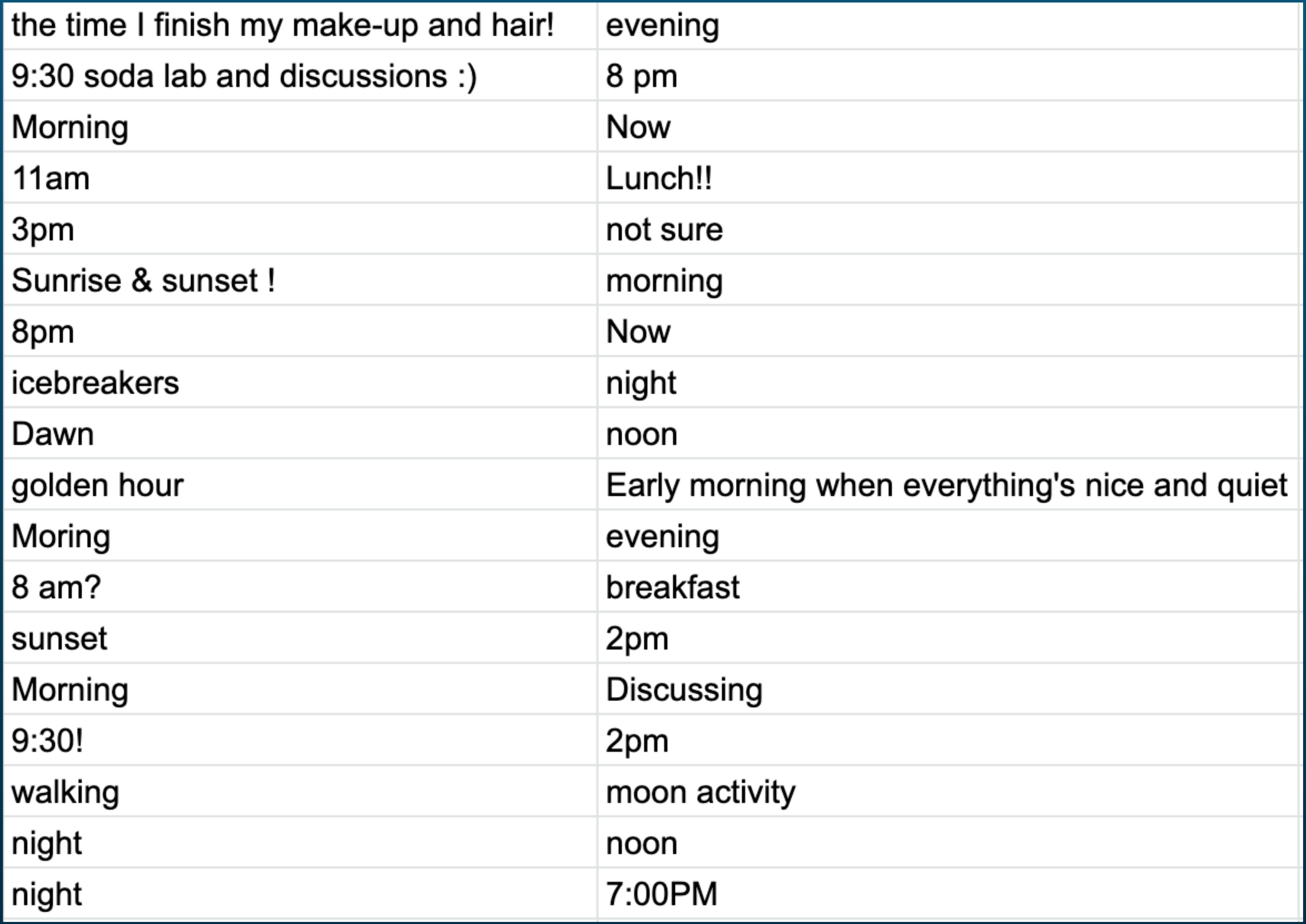
Control, Environment Diagrams

Mingxiao Wei
mingxiaowei@berkeley.edu

Jan 26, 2023

FROM LAST TIME... 👀

What's the best time of the day?



the time I finish my make-up and hair!	evening
9:30 soda lab and discussions :)	8 pm
Morning	Now
11am	Lunch!!
3pm	not sure
Sunrise & sunset !	morning
8pm	Now
icebreakers	night
Dawn	noon
golden hour	Early morning when everything's nice and quiet
Moring	evening
8 am?	breakfast
sunset	2pm
Morning	Discussing
9:30!	2pm
walking	moon activity
night	noon
night	7:00PM

LOGISTICS 🏡

- Homework 01 due today 01/26 @ 11:59pm
- About office hours:
  - We are experimenting with a new OH system this semester - basically I'll be appointed some time slots, not necessarily every week, and I still don't know when they will be yet :/
  - That being said, we do have OH throughout the week and you can check the time/location on our course calendar
- Make sure to check Ed frequently!
- When emailing me, please try to put sth like `[CS 61A]` in the title so it's easier for me to sort things out!

CONTROL STRUCTURES 🕹️

CONDITIONAL STATEMENTS

`if-elif-else` Syntax

```
if <cond_expr_1>:
    <suite_1>
elif <cond_expr_2>:
    <suite_2>
else:
    <suite_3>
```

- Conditional expressions - essentially any expression can be a conditional expression, and it evaluates to some value that's either truthy or falsy
- There can be 0 or 1 `else`
- There can be ≥ 0 `elif`
- Only the first `if` or `elif` that evaluates to a truthy value will have its corresponding indented suite be executed
- If none of the conditional expressions evaluate to a truthy value, the `else` suite is executed.

`while` LOOPS

```
while <condition>:
    <do something>
```

- While `<condition>` is truthy, execute the body of the loop. Then come back to the condition to re-evaluate it. If truthy, execute the loop body again. Continue until the condition becomes falsy.
- To avoid infinite loops, make sure that the condition is tending towards a falsy value in the body of the loop!

BOOLEAN OPERATORS 🎏

TRUTHINESS AND FALSINESS

Truthy Values

- Treated as practically "true" in boolean contexts (if/while conditions, and/or/not expressions)
- Everything that's not falsy is truthy

Falsy Values

- Treated as practically "false" in boolean contexts (if/while conditions, and/or/not expressions)
- Including: `0`, `None`, `""` (empty string), `[]` (empty list), etc.
- See here for a comprehensive list

BOOLEAN OPERATORS

- Some arithmetic expressions evaluate to boolean values

```
>>> 6 + 1 == 7
True
>>> 8 > 9
False
```

- Python boolean operators: `not`, `and`, and `or`
  - Priority: `not` > `and` > `or`
  - Use parenthesis to make your code more readable!
- `not` - returns the opposite boolean value of the following expression
  - always return either `True` or `False`

```
>>> not 0
True
>>> not -1
False
```

SHORT CIRCUITING

- Short circuit - not every operand gets evaluated
- `and`
  - evalutes from left to right until the first FALSY value or the last value
  - return the last thing that's evaluated
- `or`
  - evalutes from left to right until the first TRUTHY value or the last value
  - return the last thing that's evaluated
- If an error occurs, the execution flow is terminated immediately

```
>>> True and 1 / 0 and 2
ZeroDivisionError
>>> True or 1 / 0 or 2
True
```

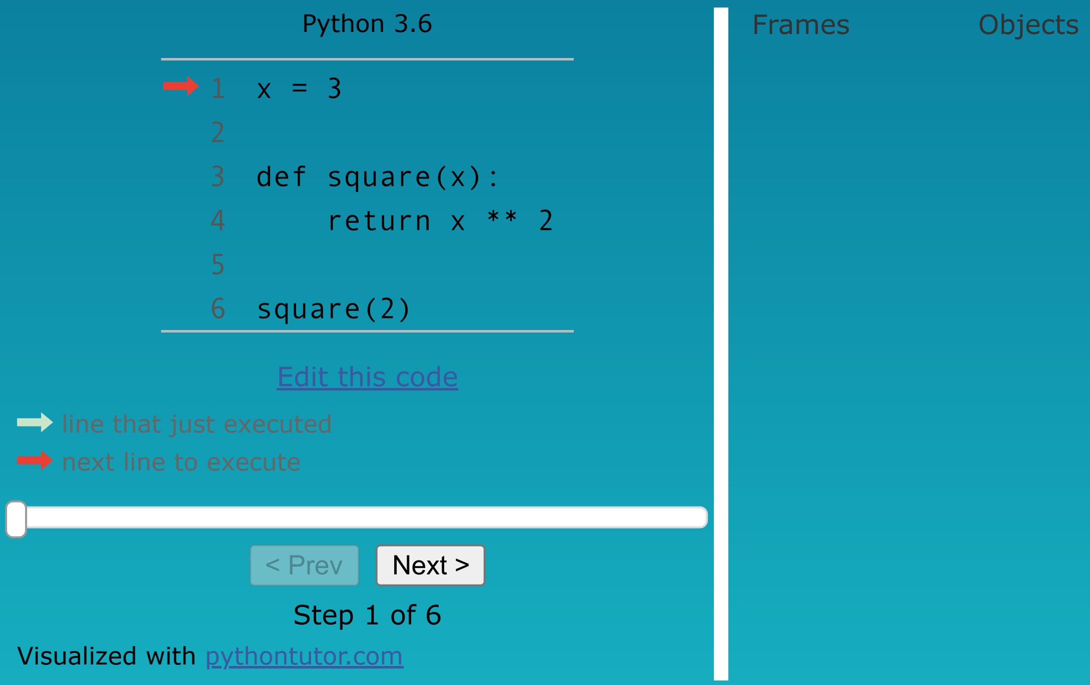
```
>>> 1 and 2 and 3
3
>>> 1 or 2 or 3
1
```

WORKSHEET Q1-6

ENVIRONMENT DIAGRAMS 🏞️

ENVIRONMENT DIAGRAMS

- A model used to keep track of variables defined and the corresponding values they are bound to
- Frame - a frame keeps track of what variables have been defined in the current execution environment
- Global frame - the frame we start off with when executing a program from scratch



```
x = 3

def square(x):
    return x ** 2

square(2)
```

ASSIGNMENT STATEMENTS

Anatomy: `<variable name> = <expression>`

- Assignment statements define variable in programs
- To execute an assignment statement:
    1. Evaluate all of the expressions on the RHS of the assignment operator (the single equal sign) from left to right
    2. Bind all the names on the LHS to the corresponding resulting values in the current frame

    * Names can only bind to values, not other names!

WORKSHEET Q7

DEF STATEMENTS

```
def <function name>:
    <function body>
    return ...
```

- A `def` statement creates a function object and binds it to a name
- In an environment diagram:
    - Write the function object to the right-hand-side of the frames, denoting the intrinsic name of the function, its parameters, and the parent frame (e.g., `func square(x) [parent = Global]`)
        - parent frame = the frame where the function is defined
    - Write the function name as a variable in the current frame and draw an arrow from the name to the function object

WORKSHEET Q8

ATTENDANCE! 🤠

go.cs61a.org/mingxiao-att

- Also linked on our section website!
- Slides: go.cs61a.org/mingxiao-index