Problem Solving and Abstraction (CMPU 101)

Tom Ellman

Lecture 1

What is Computer Science 101?

Who is qualified to succeed this course?

Who is qualified to succeed this course?

- Alive
- Human Being
- Enrolled as Student
- At Vassar

What is Computer Science?

The study of computation, i.e., using

computers,

algorithms

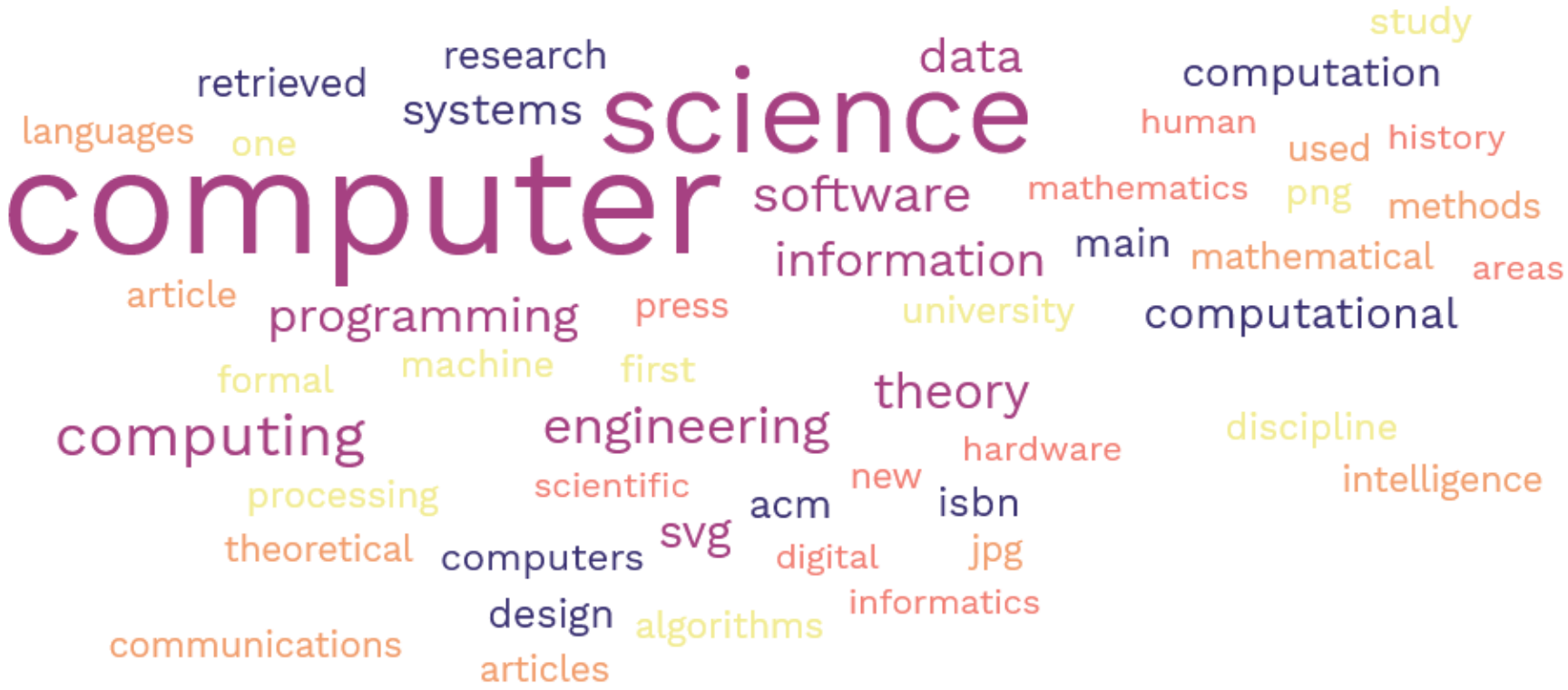
and

data-structures

to solve computational problems. (Wikipedia)

But, what are computation, algorithms, data structures and computational problems?

Word Cloud of Text on Computer Science Page on Wikipedia



Definitions

- Algorithm: A step-by-step procedure for solving a problem or performing a task: like a recipe.

- Data-Structure: A collection of data organized to facilitate solution of a computational problem.

Solving Problems with Computation

Define the problem:

- What data do we have? (Input)
- What data do we want? (Output)
- How is output related to input?

Solving (Word Cloud)Problems with Computation

Define the problem:

- What data do we have? (Input)

  Text from Computer Science Wikipedia Page

- What data do we want? (Output)

  Arrange words from the text forming a picture.

- How is output related to input?

  Word font size indicates number of occurrences in text.

Solving Problems with Computation

Design a program:

- How to represent the data in the program?

  Table recording number of occurrences of each word.

- How should the program manipulate the data?

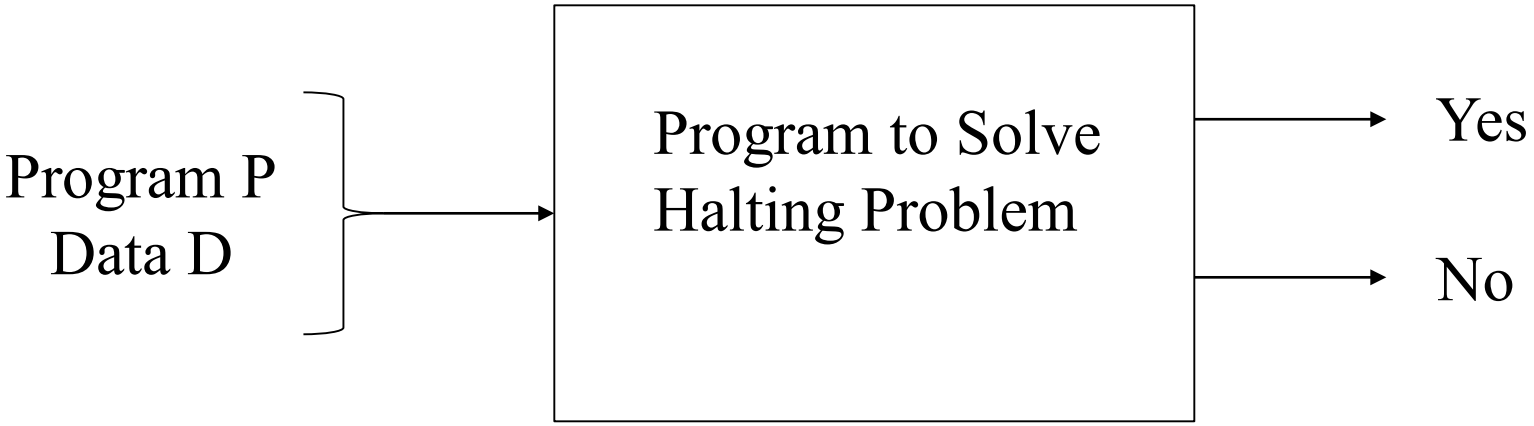
  Display each word in a random position with font size proportional to its number of occurrences.

Theoretical Computer Science

- Computability:
  - What kinds of problems are computable?
  - What kinds of problems are not computable?

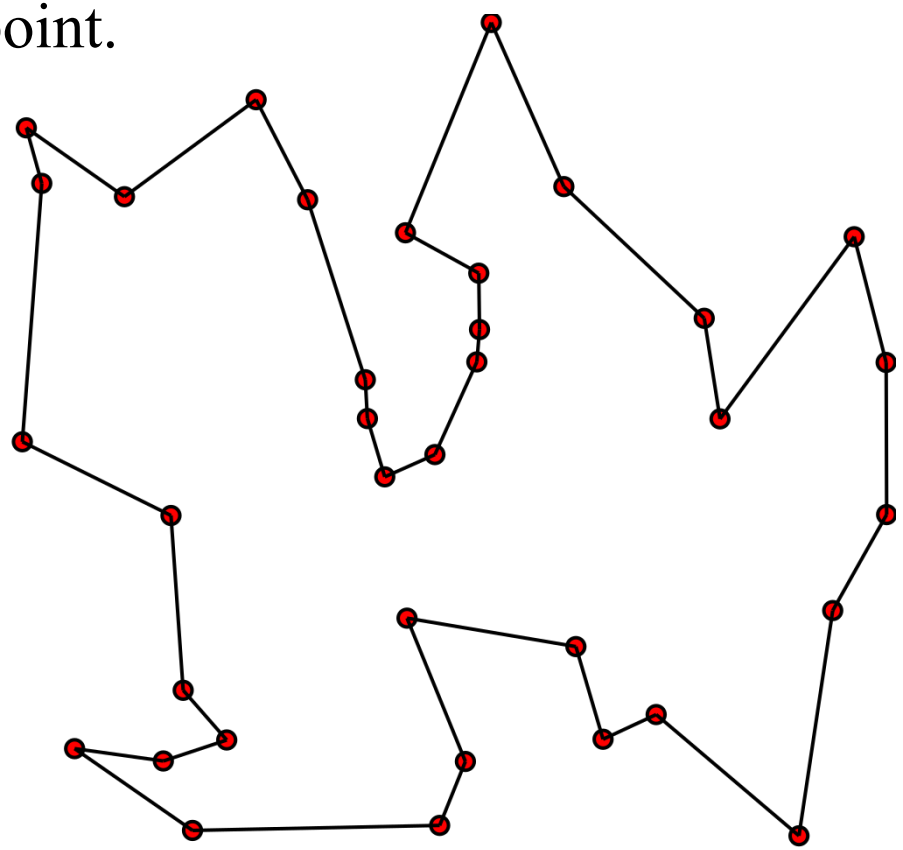
- Complexity:
  - How does running time grow as problems get larger?
  - What kinds of problem are intractable, i.e., inherently difficult to solve?

Halting Problem: If we give data *D* to program *P*, will it eventually halt, or will it run forever?



There is no program that can solve the Halting Problem for every *P* and every *D* so the Halting Problem is **not computable**.

Traveling Salesman (sic) Problem: Find the shortest path that visits each of the red dots one time and returns to the starting point.



Wikipedia

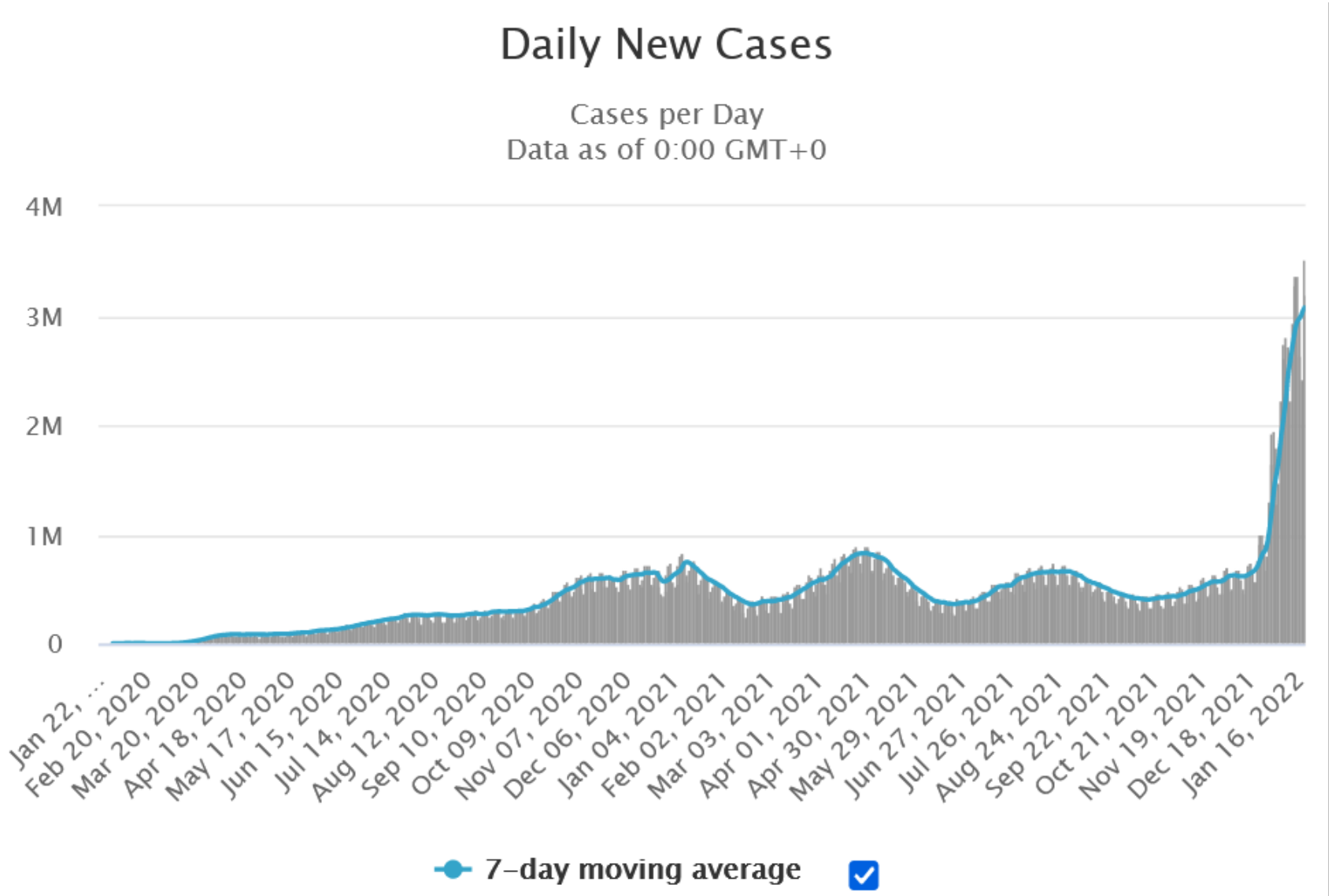
As far as we know, there is no algorithm is significantly faster than trying all possible paths to find the shortest one. P = NP ?

Data-Centric Pedagogy

- We begin by focusing on data:
  - Types of data
  - Ways of organizing data.

- Later we emphasize algorithms.
  - Data structures guide design of algorithm.
  - Algorithm guides design of data structures.

Daily New Cases of Coronavirus by Day from Worldometer.info



Police use of Amazon’s Face-Recognition Service

Class Web Site

- Visit: https://www.cs.vassar.edu/~thellman/cs101_01
- Syllabus (Read it Entirely, Soon!)
- How to Succeed in Class by Really Trying. (Read this too!)
- Links to online resources:

  CPO, Colab, GradeScope, Coaching
- Reading Assignments:

  "Data Centric Introduction to Computing", Fisler et al.
- Lecture Notes, Lab and Assignment instructions.

Contacting Tom Ellman

- Email: thellman@vassar.edu
- Office: Sanders Physics 104.2
- Telephone: (845) 437-5991
- Web Page: http://pages.vassar.edu/tomellman
- Zoom: https://vassar.zoom.us/j/5469695010
- Office Hours: MW 2pm-3pm; TR 1pm-2pm

Course Work

- Weekly Reading
- Weekly Labs
- Weekly Assignments
- Three Examinations

Lateness Policy

- Modest penalties for late homework.
- If you don’t finish by the deadline, you can submit your work by the late deadline i.e., the start of the next class meeting.
- If you miss the official deadline, but meet the late deadline you will receive a 10% grade penalty.

Grading

- Assignments: 20%
- Labs: 20%
- Exam 1: 20%
- Exam 2: 20%
- Exam 3: 20%

Getting Help

- Instructor: Office Hours & Email
- Coaches: Office Hours & Email

Collaboration Policy for Assignments

- Don’t look at other student’s code.
- Don’t discuss code with another student.
- Don’t discuss pseudo-code with another student.
- Do discuss approaches to problems.
- These restrictions do not apply to lab work.

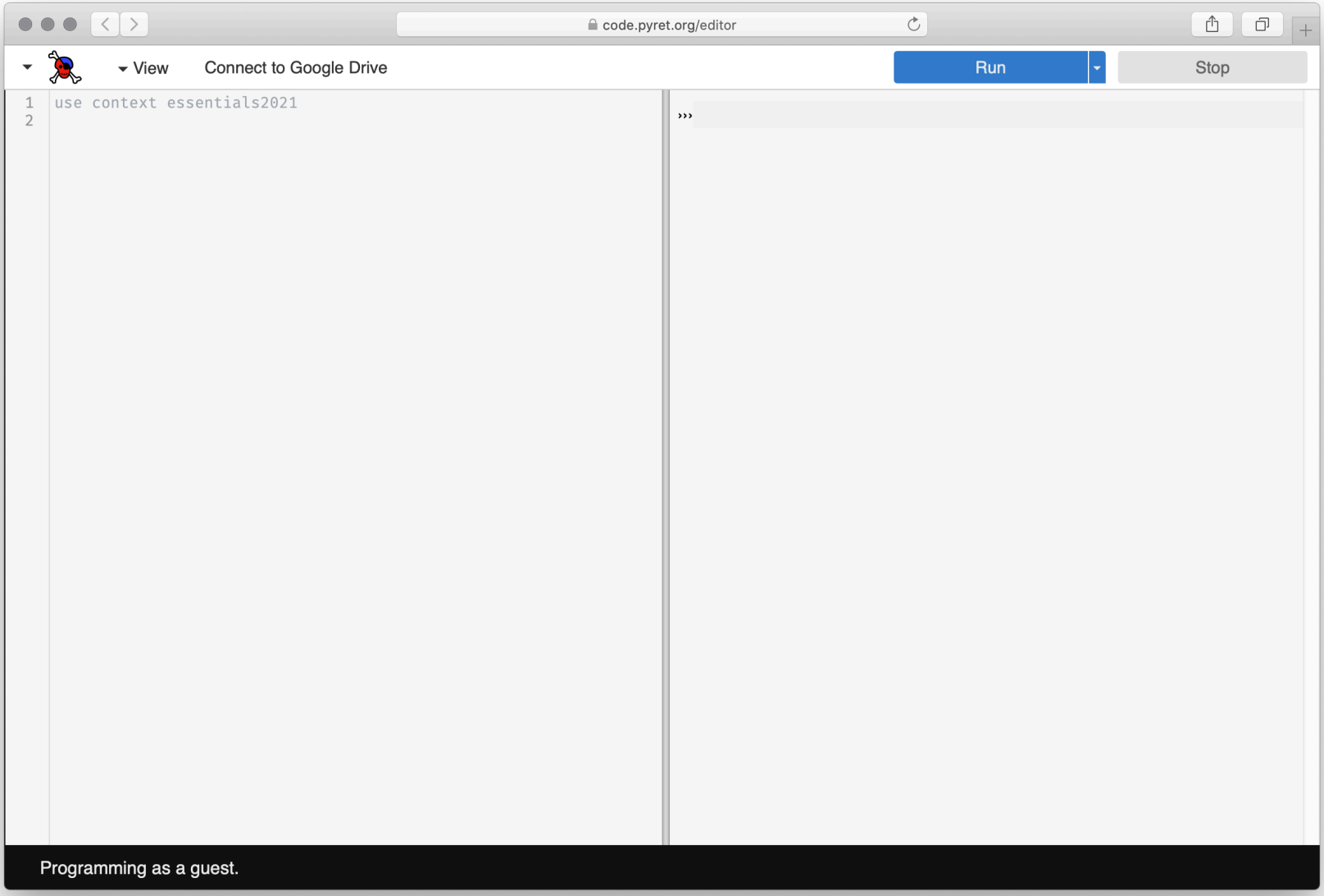
Academic Integrity

- Read the Computer Science department's Guide to Academic Integrity.

- Also consult the Syllabus. It describes some special rules that that you must follow in this course.

Languages and Platforms

- Pyret:
  - Runs in your web browser.
  - CPO (code.pyret.org)
- Python:
  - Also runs in your browser.
  - Jupyter Notebooks.
- Linux: Operating system that runs on the machines in our computer labs.



Definitions

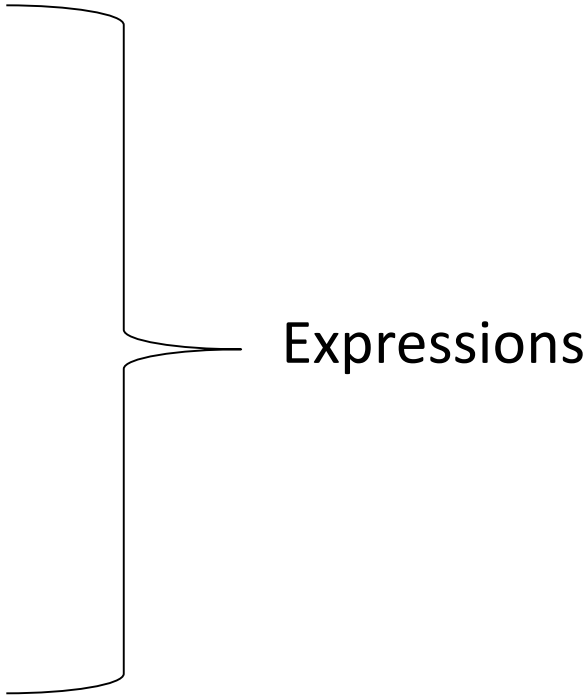
Interactions

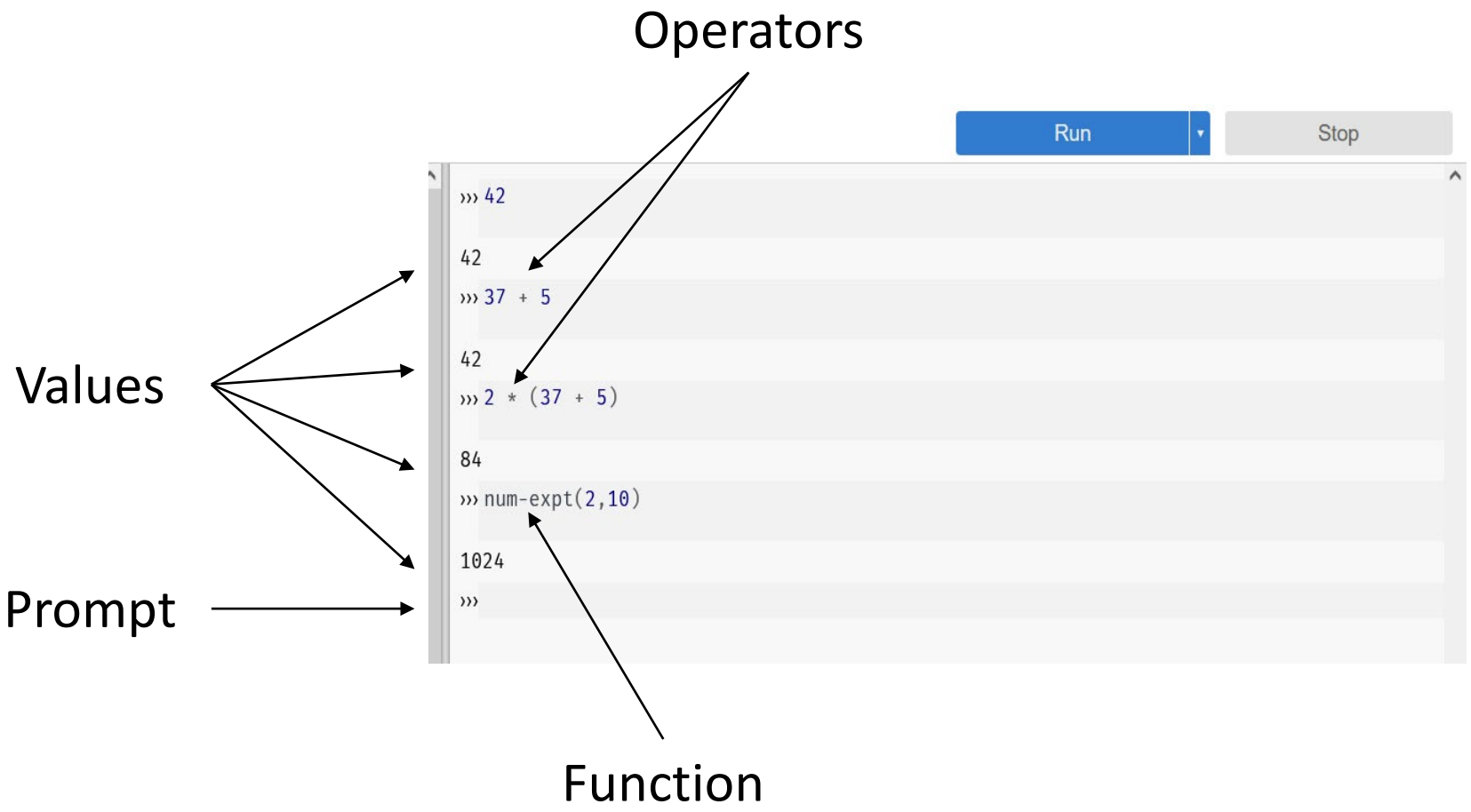
CPO's Read-Evaluate-Print Loop

- Read the expression typed by the user.
- Evaluate the expression, i.e., simplify the expression until it cannot be simpler.
- Print the value obtained by simplification.

Try these expressions!

```
42

37 + 5

2 * (37 + 5)

num-expt(2,10)
```




Operators
Run
Stop
››› 42
42
››› 37 + 5
42
››› 2 * (37 + 5)
84
››› num-expt(2,10)
1024
›››
Values
Prompt
Function

Try these expressions!

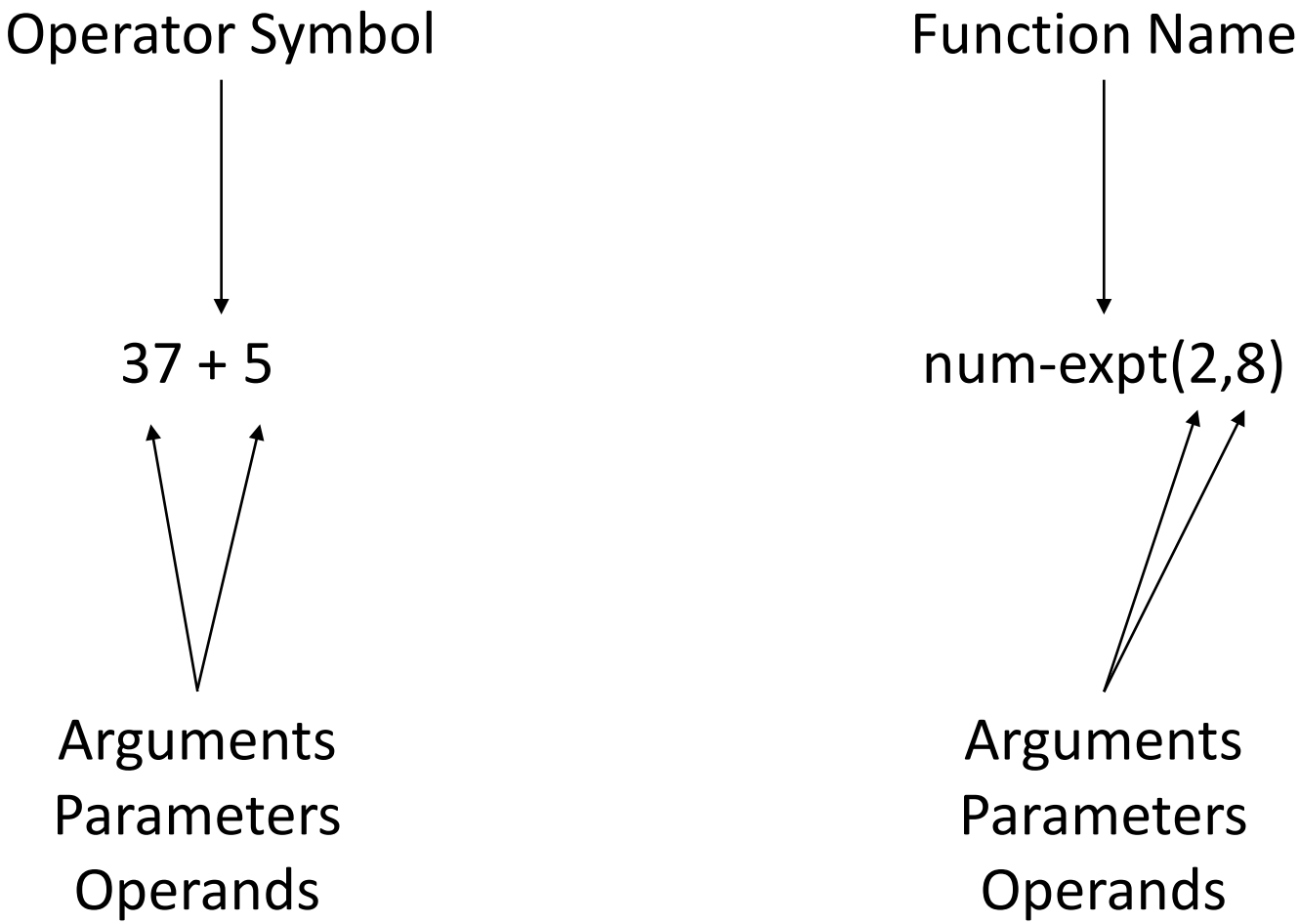
```
fish

37+5

2 * 37 + 5

num-expt(2)
```

Read the error messages. Are they helpful? Yes, sometimes!

Expressions vs. Values

- An expression describes a computation.
- Expressions describe computations in terms of values and operators / functions.
- A value is an expression that cannot be further simplified by carrying out more computations.

Functions and Operators



Evaluation is the process of converting an expression into a value, using operators and functions.

Evaluation is a kind of computation.

Computation

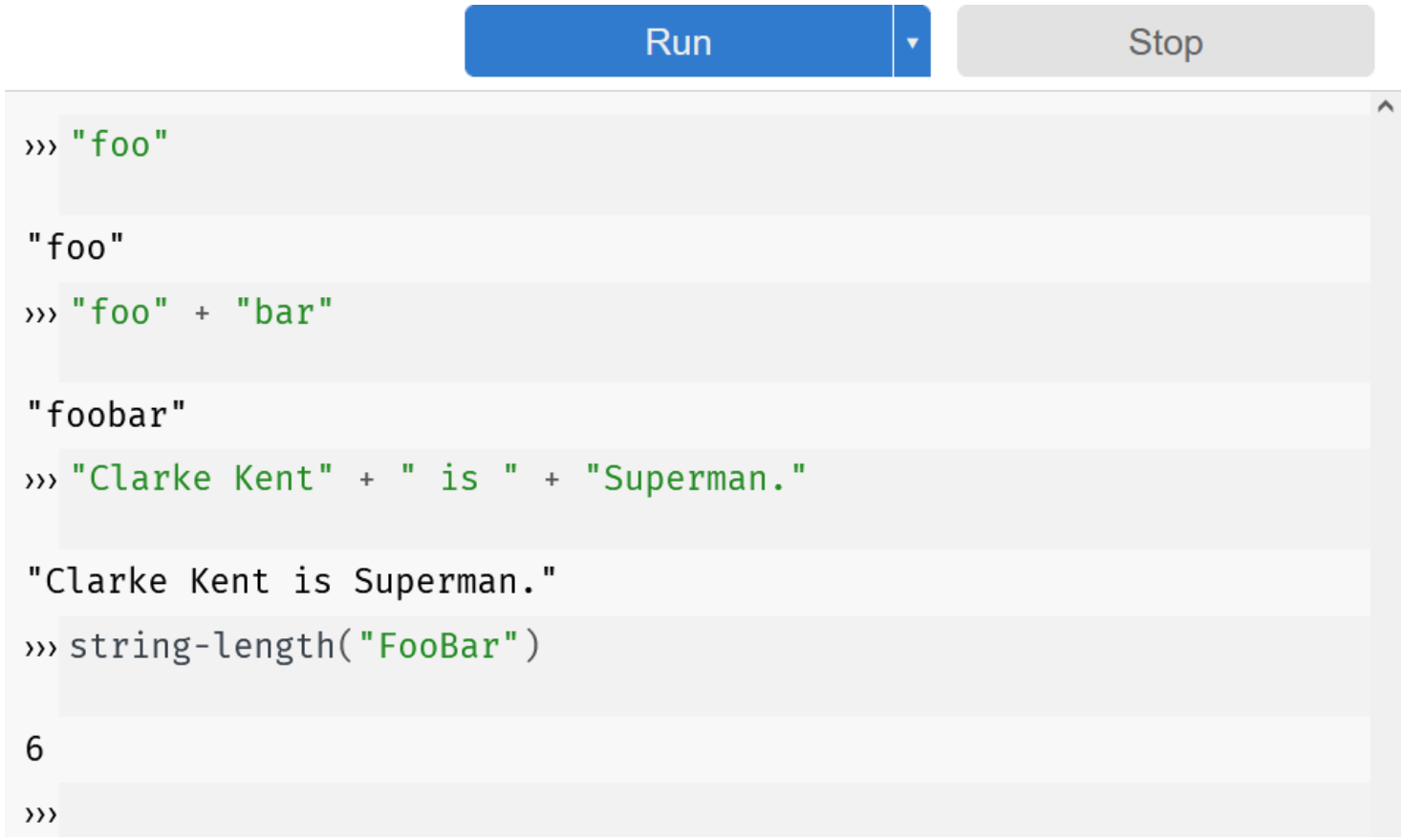
- A series of numerical calculations or symbol manipulations.
- Numerical calculations include:
  - Arithmetic operations (add, subtract, multiply, divide).
  - Trigonometric functions (sine, cosine, tangent).
  - ...
- Symbolic manipulations include:
  - Comparison of numbers or symbols.
  - Decisions of what instructions to do next.
  - Substitutions of one string of letters and numbers for another.

Types of Data

- Number
- String
- Image
- …

Try these expressions!

```
"foo"

"foo" + "bar"

"Clarke Kent"+ " is " + "Superman."

string-length("FooBar")
```


Run
Stop
››› "foo"
"foo"
››› "foo" + "bar"
"foobar"
››› "Clarke Kent" + " is " + "Superman."
"Clarke Kent is Superman."
››› string-length("FooBar")
6
›››

String

- A sequence of characters including letters, numbers and punctuation, and more.

- The character sequence must be enclosed in quotes: "fish" or 'fish'.

Try these expressions!

```
"Clarke Kent

Superman"

"Lois Lane"   "Jimmy Olsen"

3 * "FooBar"
```

Read the error messages. Are they helpful? Yes, sometimes!

Kinds of (Syntax) Errors

- No space on sides of operators.
- Wrong number of data given to a function.
- Wrong type of data given to an operator or function.
- Expression lacking parentheses.
- Mismatched parentheses.
- String missing quotes.
- Mismatched quotes.
- Multiple complete expressions on one line.
- Use of undefined names.

Definitions

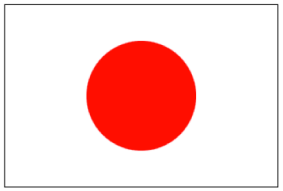
- Expression vs. Value
- Data vs. Data Type
- Operator, Function

Armenia

Austria

Colombia

Zambia



Japan

How are flags similar to each other?
How are they different from each other?
What building blocks would be useful?

Try these expressions!

```
include image #Loads the image library.

circle(25, "solid", "green")

rectangle(20, 40, "solid", "blue")

triangle(30, "solid", "red")

above(triangle(30, "solid", "red"),
      rectangle(20, 40, "solid", "blue"))
```

```
››› include image #Loads the image library.
››› circle(25, "solid", "green")
```

```
››› rectangle(20, 40, "solid", "blue")
```

```
››› triangle(30, "solid", "red")
```

```
››› above(triangle(30, "solid", "red"),
    rectangle(20, 40, "solid", "blue"))
```

```
›››
```

Everything after the pound (#) sign is a comment, for humans only. Pyret ignores it.

Names

Run | Stop

```
>>> x = 17
>>> y = 3
>>> x + y
20
>>> title = "President"
>>> sir-name = "Bradley"
>>> title + " " + sir-name
"President Bradley"
>>>
```

Try These Expressions!

```
include image
r1 = rectangle(20, 40, "solid", "blue")
r1
t1 = triangle(30, "solid", "red")
t1
house = above(t1, r1)
house
beside(house, house)
```

```
>>> include image
>>> r1 = rectangle(20, 40, "solid", "blue")
>>> r1
```

```
>>> t1 = triangle(30, "solid", "red")
>>> t1
```

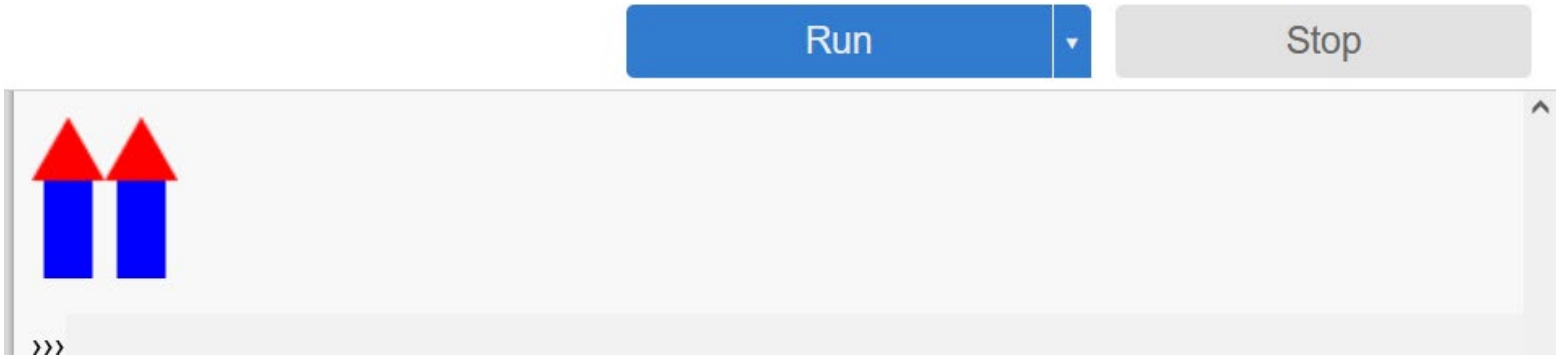
```
>>> house = above(t1, r1)
>>> house
```

```
>>> beside(house, house)
```

Cut and Paste into the Definitions Pane & Press **Run**

```
include image
r1 = rectangle(20, 40, "solid", "blue")
t1 = triangle(30, "solid", "red")
house = above(t1, r1)
beside(house, house)
```

Definitions



Interactions

Using Documentation

- Click LMB on Pirate Icon.
- Select Documentation menu item.
- Enter "image" (without quotes) in the search bar.
- Click on "Basic Images".
- Look at Circle documentation.
- Click on "FillMode" and "ImageColor".


How do you draw so well?
Practice.
It must be an innate gift...
A gift from God...
It's practice.
I'll never understand how some people are so talented...
A mystery...
>
Practice.
©Sarah Andersen