

Testing & Debugging

DEFENSIVE PROGRAMMING

- Write **specifications** for functions
- **Modularize** programs
- Check **conditions** on inputs/outputs (assertions)

TESTING/VALIDATION

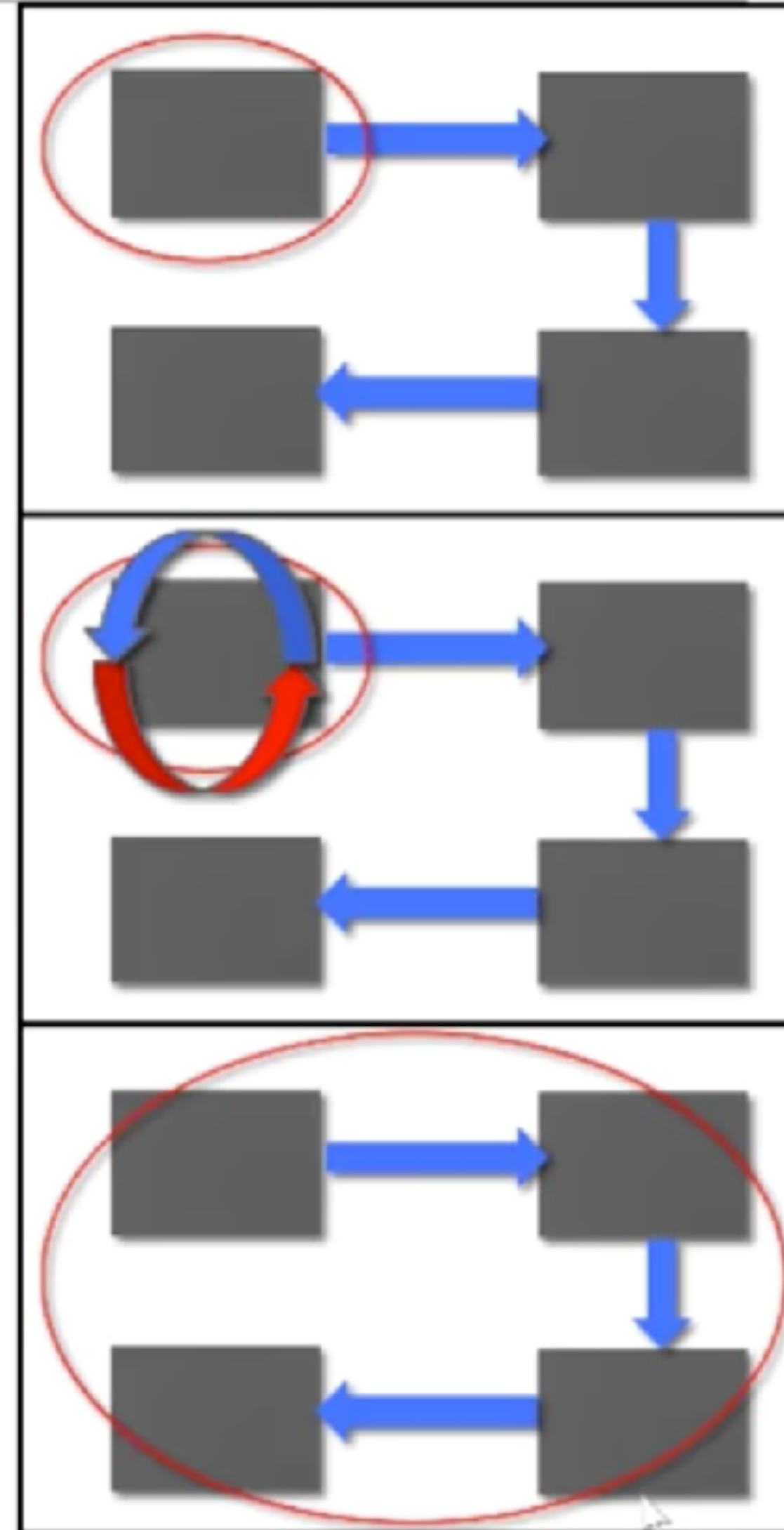
- **Compare** input/output pairs to specification
- “It’s not working!”
- “How can I break my program?”

DEBUGGING

- **Study events** leading up to an error
- “Why is it not working?”
- “How can I fix my program?”

CLASSES OF TESTS

- **Unit testing**
 - validate each piece of program
 - **testing each function** separately
- **Regression testing**
 - add test for bugs as you find them in a function
 - **catch reintroduced** errors that were previously fixed
- **Integration testing**
 - does **overall program** work?
 - tend to rush to do this



TESTING APPROACHES

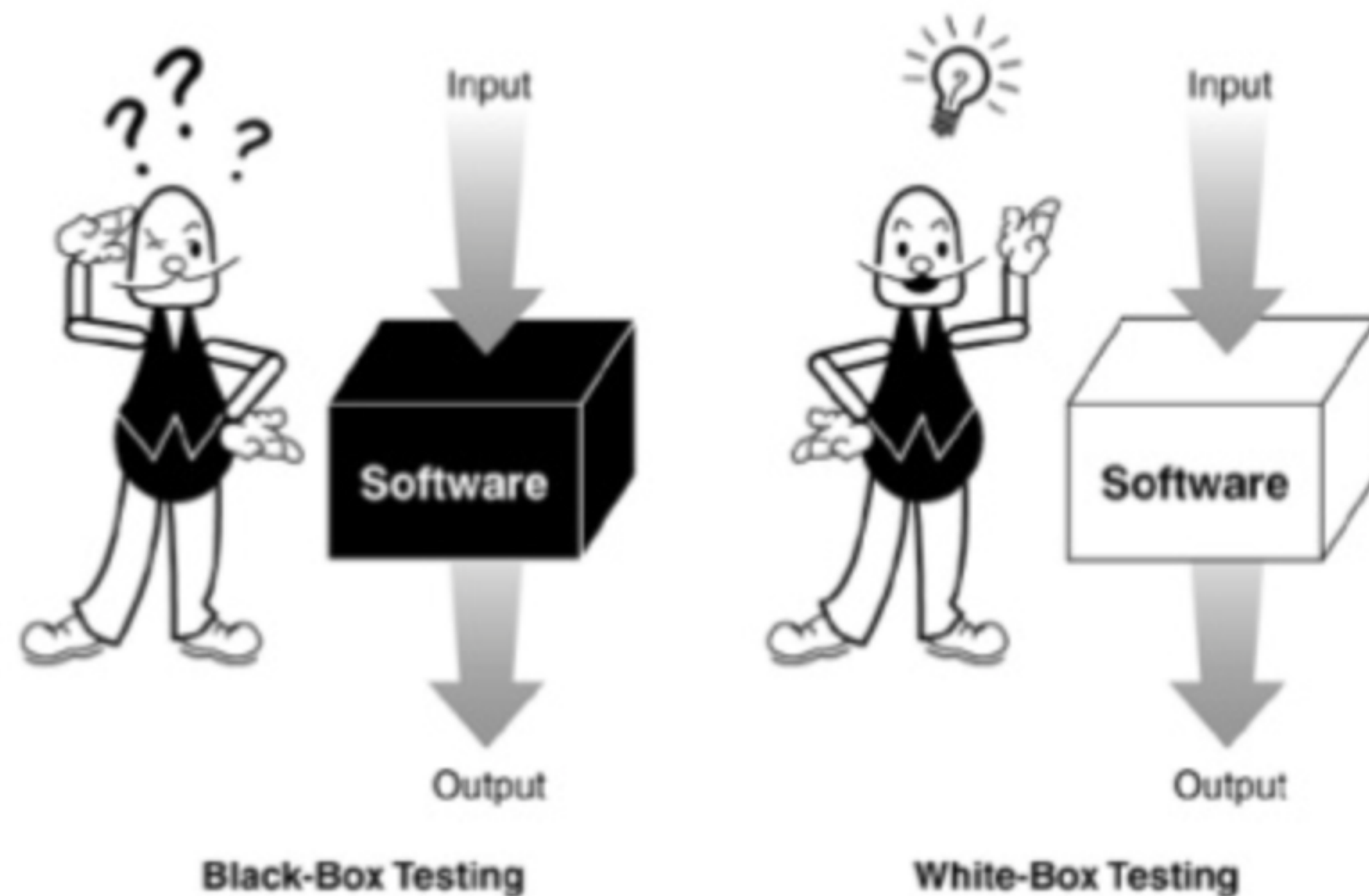
- **intuition** about natural boundaries to the problem

```
def is_bigger(x, y):  
    """ Assumes x and y are ints  
    Returns True if y is less than x, else False """
```

- can you come up with some natural partitions?
- if no natural partitions, might do **random testing**
 - probability that code is correct increases with more tests
 - better options below
- **black box testing**
 - explore paths through specification
- **glass box testing**
 - explore paths through code

BLACK-BOX VS. WHITE-BOX

- Test cases derived from specifications



- The focus is not the design, nor the implementation
 - The focus is on the logic of implementation



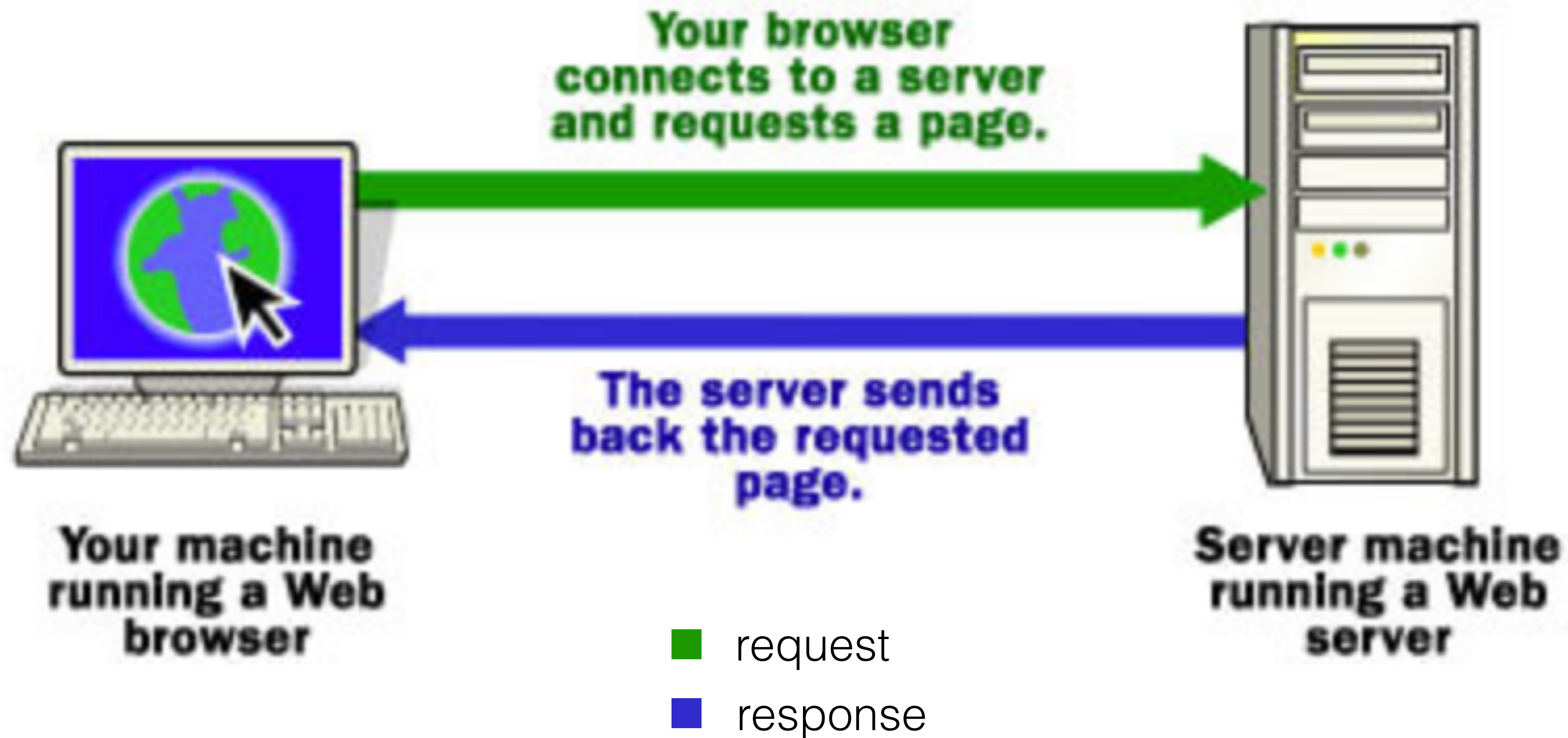
Debugging is a search problem

Tips

- preguntarse porque el código esta haciendo lo que hace, en vez de preguntarse porque no hace lo que quiero que haga.
- el 'bug' probablemente no este donde uno cree, ir descartando posibilidades
- hablar sobre el problema con alguien más
- no creer la documentación al 100%
- tomarse un descanso

cada dato en internet esta almacenado en
un servidor remoto - una computadora
optimizada para procesar *peticiones*

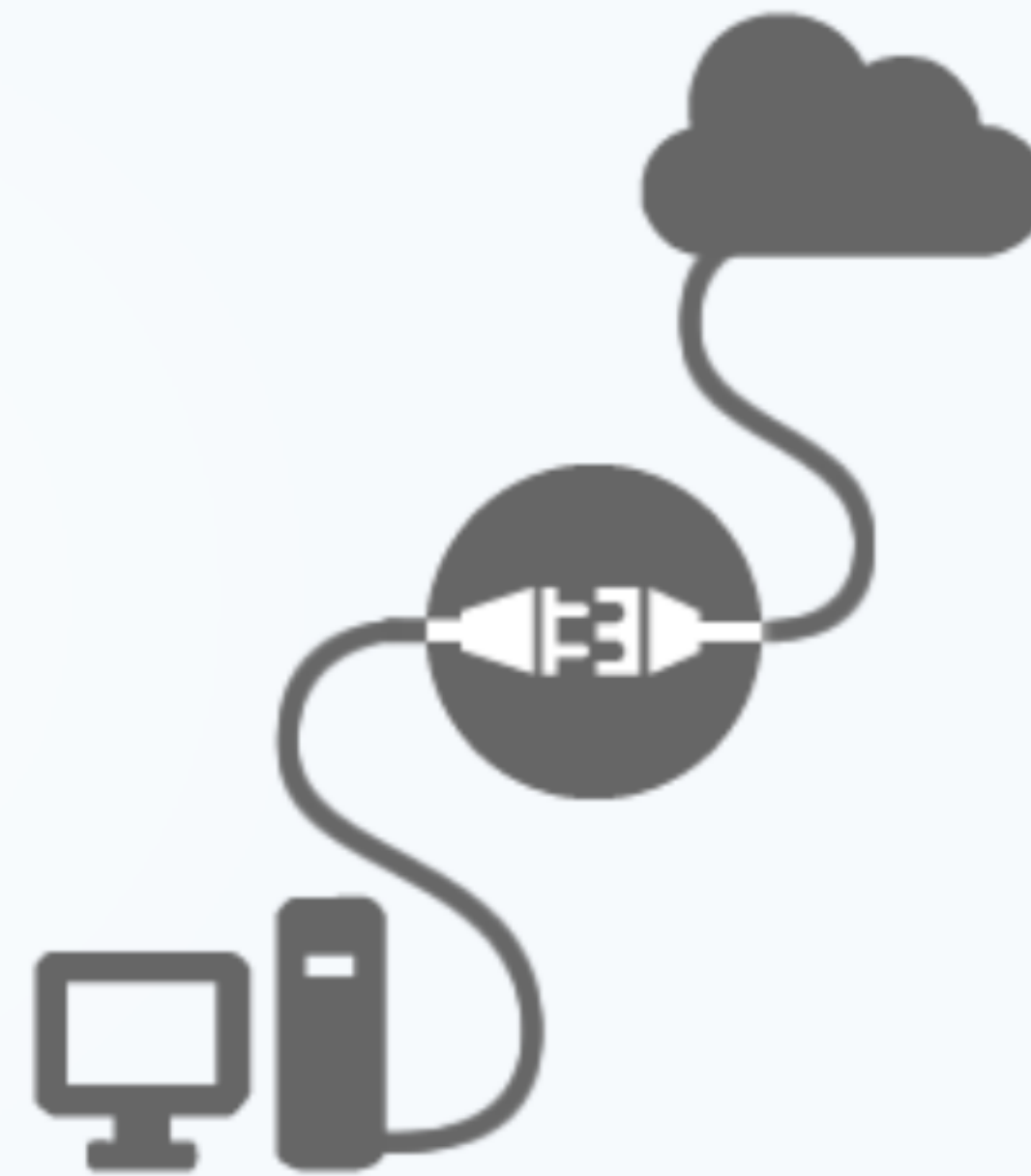
remote server production



Web Scrapping vs. Web APIs



VS



Web Scrapping

- Todos los están hechos de HTML, y HTML = data.
- Empresas dedican más mantenimiento a su front end.
- No hay rate-limiting, pero captchas.
- Es “anonimo”. [IP, cookies]
- Puede ser ilegal.

Web APIs

- Application Programming Interface.
- No es lo mismo que un servidor remoto, más bien es la parte del servidor remoto encargada de procesar *peticiones* y sus *respuestas*.
- Servicio al cliente.
- Third-party APIs y microservices.
- No GUI, solo data. [En vez de HTML, JSON.]
- Autenticación y documentación.

local server development

Your browser connects to
your local server



And Displays the
Data in the browser

Your machine running a
Web Browser & a Web Server

- request
- response