Asynchronous JavaScript

Ryan Lewis
rylewis@expedia.com

node JS 101

Class Overview

Callbacks

Anonymous Functions

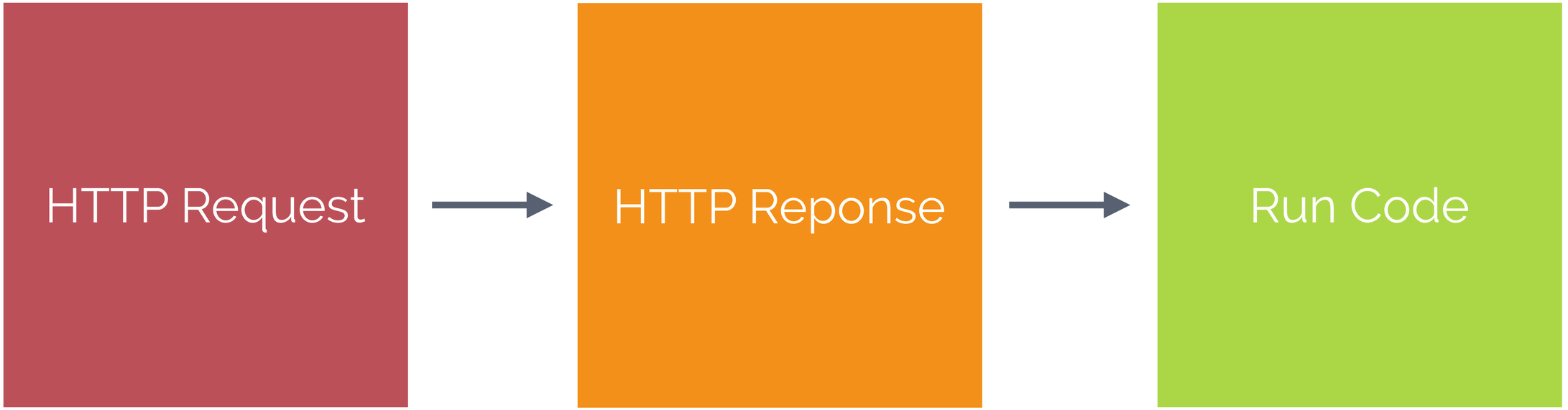
homework solution

Slow Computing Operations

I/O

Network Requests

Databases


HTTP Request
HTTP Reponse
Run Code


JavaScript
downloaded from pickywallpapers.com

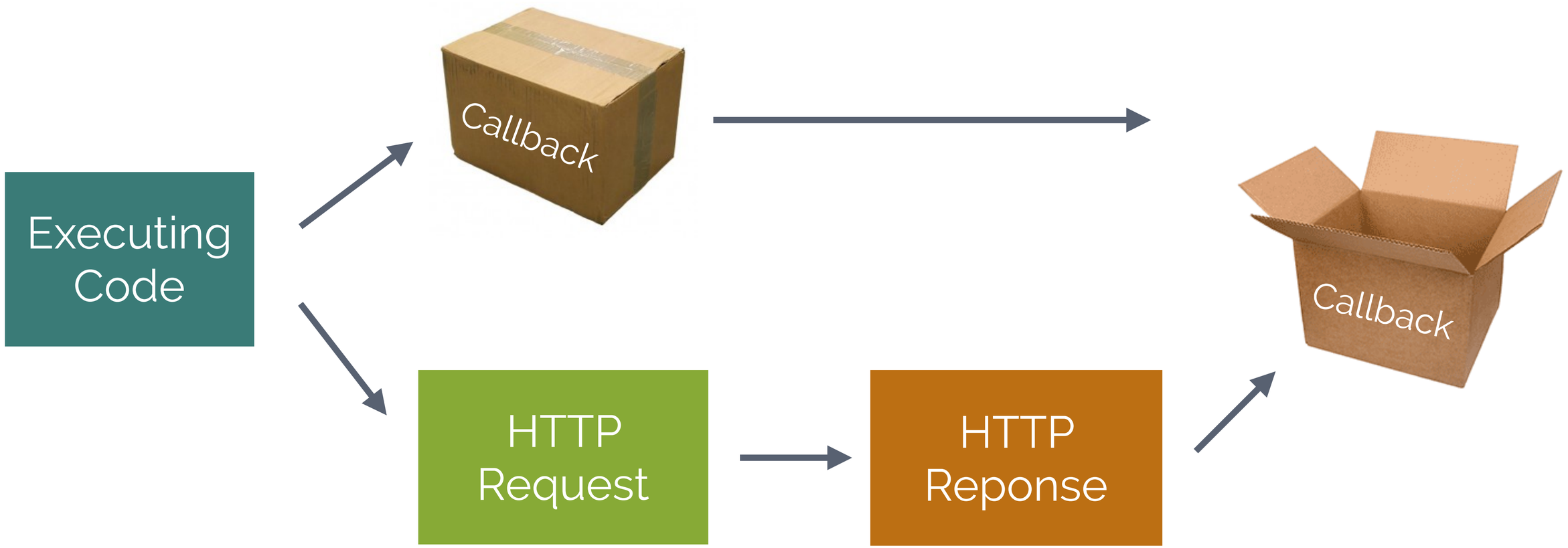
```
var request = require(‘request’);

var result = request.get(‘https://www.google.com');

console.log(results);
console.log(‘done’);


// undefined
// ‘done’
```

Super
Callback!


Executing Code
Callback
HTTP Request
HTTP Reponse
Callback

```
var request = require('request');

function reqCallback(err, res, body) {
  console.log(body);
  console.log('done');
}

request.get('https://www.google.com', reqCallback);
```

When To Use Callback Pattern

Async Operations

Return Multiple Arguments

Node ‘Failback’ Pattern

```
function myCallback (err, data) { }
```

Error First
*if falsey, no errors

Returned data
or parameters

***Note: Callback function signature decided by *calling* function

Anonymous Function

Function that has not been assigned to a variable. Usually passed as an argument to another function.

```
var request = require('request');

request.get('https://www.google.com', function(err,
res, body) {
  console.log(body);
  console.log('done');
});
```

Why?

Quick to write

No unnecessary variables

Why Not?

No reference in stacktraces

Can’t reuse

Can be difficult to read