ASYNC PROGRAMMING

What does this print?

```
function getY() {
   var y;
   $http.get("/gety", function(jsonData){
    // suppose the value of y on the server is 3
     y = jsonData.y;
   });
   return y;
}

var x = 5;
var y = getY();

console.log(x + y);
```

Can't return values in async world!

```
function getY() {
   var y;
   $http.get("/gety", function(jsonData){
      y = jsonData.y;
   });
   return y;
}

var x = 5;
var y = getY();

console.log(x + y);
```

Continuation Passing Style

CONTINUATION PASSING STYLE (CPS)

```
function getY(continueWith) {
  $http.get("/gety", function(jsonData) {
    continueWith(jsonData.y);
  });
}

var x = 5;

getY(function(y) {
  console.log(x + y);
});
```

CALLBACK STYLE PROGRAMMING

```
fs.readdir(source, function(err, files) {
  if (err) {
    console.log('Error finding files: ' + err)
  } else {
    files.forEach(function(filename, fileIndex) {
      console.log(filename)
      gm(source + filename).size(function(err, values) {
        if (err) {
          console.log('Error identifying file size: ' + err)
        } else {
          console.log(filename + ' : ' + values)
          aspect = (values.width / values.height)
          widths.forEach(function(width, widthIndex) {
            height = Math.round(width / aspect)
            console.log('resizing ' + filename + 'to ' + height + 'x' + height)
            this.resize(width, height).write(destination + 'w' + width + '_' + filename, function(err)
              if (err) console.log('Error writing file: ' + err)
            })
          }.bind(this))
        }
      })
    })
  }
})
```

callbackhell.com

Promises

PROMISES

software abstraction for dealing with “callback hell”

move from CPS style

```
getTweetsFor("domenic", function (err, results) {
    // the rest of your code goes here.
});
```

to one where functions return a value, called a promise

```
var promiseForTweets = getTweetsFor("domenic");
```

https://gist.github.com/domenic/3889970

CommonJS Promises/A

A promise is defined as an object that has a function as the value for the property 'then':

```
then(fulfilledHandler, errorHandler, progressHandler)
```

Adds a **`fulfilledHandler`**, **`errorHandler`**, and **`progressHandler`** to be called for completion of a promise. The **`fulfilledHandler`** is called when the promise is fulfilled. The **`errorHandler`** is called when a promise fails. The **`progressHandler`** is called for progress events. All arguments are optional and non-function values are ignored…

wiki.commonjs.org/wiki/Promises/A

CommonJS Promises/A

This function **should return a new promise** that is fulfilled when the given fulfilledHandler or errorHandler callback is finished. **This allows promise operations to be chained together.** The value returned from the callback handler is the fulfillment value for the returned promise. If the callback throws an error, the returned promise will be moved to failed state.

wiki.commonjs.org/wiki/Promises/A

IMPLICATIONS

treat promises are first-class object: pass as parameters, aggregate, etc.

no more nested callbacks (CPS style)

“The point of promises is to give us back functional composition and error bubbling in the async world”

https://gist.github.com/domenic/3889970

Promise Chaining

```
getTweetsFor("domenic") // promise-returning async function
    .then(function (tweets) {
        var shortUrls = parseTweetsForUrls(tweets);
        var mostRecentShortUrl = shortUrls[0];
        return expandUrlUsingTwitterApi(mostRecentShortUrl); // promise-
returning async function
    })
    .then(doHttpRequest) // promise-returning async function
    .then(
        function (responseBody) {
            console.log("Most recent link text:", responseBody);
        },
        function (error) {
            console.error("Error with the twitterverse:", error);
        }
    );
```

Web Apis

REST vs SOAP

resources vs operations

REST new-hotness

SOAP security, ACID transactions, reliable messaging

spf13.com/post/soap-vs-rest

WEB APIs

application program interface to a defined request-response message system between clients and servers

accessible via standard HTTP methods

request URLs that transfer representations (JSON, XML)

spf13.com/post/soap-vs-rest

XMLHttpRequest

most widely deployed API client in the world

a copy in every web browser

most sites today are built on top of APIs designed for consumption by XMLHttpRequest

arRESTed Development

SEMANTIC CHALLENGE

Learning one API doesn’t help a client learn the next one

ProgrammableWeb: the world's largest API repository, GROWING DAILY

Search Over 13,068 APIs Search APIs

Filter APIs

By Category By Protocols/Formats Include Deprecated APIs

API Name	Description	Category	Updated
Google Maps	The Google Maps API allow for the embedding of Google Maps onto web pages of outside developers, using a simple JavaScript interface or a Flash interface. It is designed to work on both mobile...	Mapping	12.05.2005
Twitter	The Twitter micro-blogging service includes two RESTful APIs. The Twitter REST API methods allow developers to access core Twitter data. This includes update timelines, status data, and user...	Social	12.08.2006
YouTube	The Data API allows users to integrate their program with YouTube and allow it to perform many of the operations available on the website. It provides the capability to search for videos, retrieve...	Video	02.08.2006
Flickr	The Flickr API can be used to retrieve photos from the Flickr photo sharing service using a variety of	Photos	09.04.2005

API Directory Search

Search over 13,068 APIs updated daily

Search APIs, mashups, developers

Browse by Category

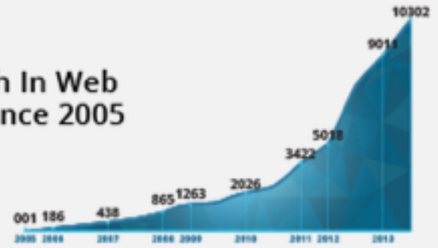
Newest APIs

Latest Mashups

Add an API +

PW Research Center

Our data. Your PowerPoints. Use our API research for your next presentation. See all →



NEXT CLASS: DATABASES

courses.engr.illinois.edu/cs498rk1/