IMA LOWRES

Overview of APIs & the Web

Connections Lab

Fall 2021

What is an **API** ?

Application **P**rogramming **I**nterface

A set of requirements that govern how one application can communicate with another

(instructions that allow computer programs to talk to each other)


Joe's PIZZ
www.joespizzanyc.com
Joe's PIZZA
FAMOUS SINCE
SLICE 2.75
SICILIAN 3.00
FRESH MOZZ. 3.50
ANY TOPPING 3.50
LARGE CHEESE 20.00
ANY TOPPING 24.00
SODA 2.00
WATER 1.50
212-366-1182 • 212-255-3946

DEVICE API - access sensor data on phone to affect content

OS API - cut and paste from a Text Editor to an Email Editor

PLATFORM API - leverage the fetch API in a web browser

FRAMEWORK API - use p5.js functions to run javascript

SERVICE API - send a text to an email via Twilio

DATA API - request a list of articles from Wikipedia

RESOURCE API - embed a Google map on a web page

API FOR APIs - use an SDK to access 100+ APIs

CODE that extends what we can do with javascript.

PLATFORM API - leverage the fetch API in a web browser

FRAMEWORK API - use p5.js functions to run javascript

SERVICE API - send a text to an email via Twilio

DATA API - request a list of articles from Wikipedia

RESOURCE API - embed a Google map on a web page

URLs that allow us to request & retrieve info from external sources

Code as API

```
fetch(url)
.then(response => response.json())
.then(data => {
  //do something with the data
  console.log(data)
})
```

AJAX

Asynchronous Javascript XML

```
let req = new XMLHttpRequest();
req.open("GET",url);
req.send();
req.addEventListener("load",function(){
  console.log(this.response);
});
```

```
$.ajax(url).done(function(data){
    console.log(data);
});
```

```
$.ajax({
  url: “http://data.com/data.json",
  type: “GET”,
  dataType: “jsonp”,
  error: function(err){
            console.log(“Uh oh”);
            console.log(err);
  },
  success: function(data){
            console.log(“WooHoo”);
            console.log(data);
  }
});
```

p5.js

Home
Editor
Download
Donate
Get Started
Reference
Libraries
Learn
Examples
Books
Community
Showcase
Forum
GitHub
Twitter

Reference

Search reference

Can't find what you're looking for? You may want to check out p5.sound.
You can also download an offline version of the reference.

Color

Creating & Reading

alpha()
blue()
brightness()
color()
green()
hue()
lerpColor()
lightness()
red()
saturation()
p5.Color

Setting

background()
clear()
colorMode()
fill()
noFill()
noStroke()
stroke()
erase()
noErase()

Shape

2D Primitives

arc()
ellipse()
circle()
line()
point()
quad()
rect()
square()
triangle()

Attributes

ellipseMode()
noSmooth()
rectMode()
smooth()
strokeCap()
strokeJoin()
strokeWeight()

Curves

bezier()
bezierDetail()
bezierPoint()
bezierTangent()
curve()
curveDetail()
curveTightness()
curvePoint()
curveTangent()

ellipse()

Examples

```
ellipse(56, 46, 55, 55);
```

edit reset copy

Description

Draws an ellipse (oval) to the screen. By default, the first two parameters set the location of the center of the ellipse, and the third and fourth parameters set the shape's width and height. If no height is specified, the value of width is used for both the width and height. If a negative height or width is specified, the absolute value is taken.

An ellipse with equal width and height is a circle.The origin may be changed with the `ellipseMode()` function.

Syntax

```
ellipse(x, y, w, [h])
```

```
ellipse(x, y, w, h, detail)
```

Parameters

x	Number: x-coordinate of the center of ellipse.
y	Number: y-coordinate of the center of ellipse.
w	Number: width of the ellipse.
h	Number: height of the ellipse. (Optional)
detail	Integer: number of radial sectors to draw (for WebGL mode)

URL as API

URLs that give access to data and resources from a web server

REQUEST

RESPONSE

CLIENT

SERVER

HTTP

HyperText Transfer Protocol

a **GET** request

`http://nyu.edu`

a response

CLIENT

SERVER

`index.html`

`style.css`
`scripts.js`

`photos.jpg`
`media.mp4`
`data.json`

a **GET** request

`http://nyu.edu`

a response

CLIENT

SERVER

index.html

style.css
scripts.js

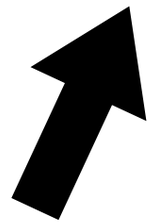
photos.jpg
media.mp4
data.json

http://api.nyu.edu/lowres/courses

`http://api.nyu.edu/lowres/courses`

Network Protocol

http://**api.nyu.edu**/lowres/courses

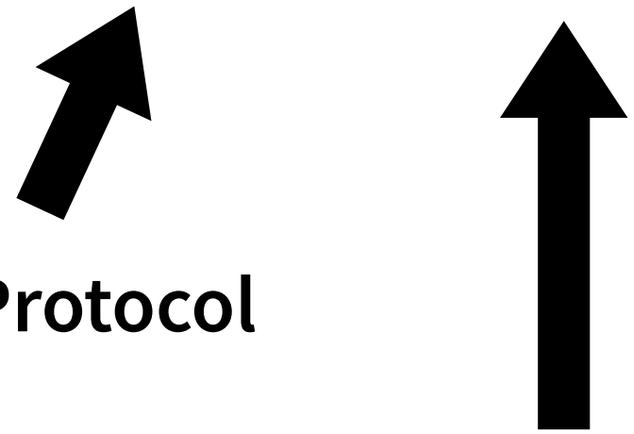


Network Protocol

Root/Host Name/Address

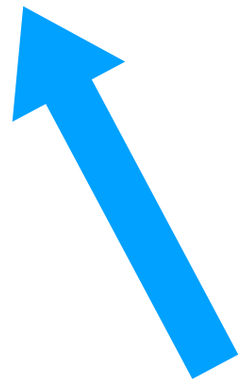
(IP - DNS)

http://api.nyu.edu/lowres/courses



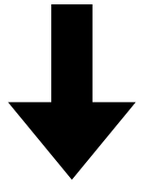
Network Protocol

Root/Host Name/Address
(IP - DNS)



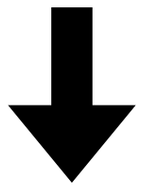
Resource Location
route/path/endpoint/uri

`http://api.nyu.edu/lowres/courses`



{ JSON DATA }

http://api.nyu.edu/lowres/courses



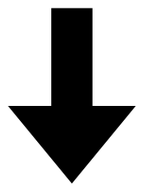

```
{
  { “code”: 101,
    “title”: “Concepts, Culture & Critique”,
  },
  { “code”: 102,
    “title”: “Creative Coding”,
  },
  { “code”: 103,
    “title”: “Design for Communication”,
  },
  { “code”: 104,
    “title”: “Interface Lab”,
  },
  { “code”: 105,
    “title”: “Conversations”,
  },
  { “code”: 201,
    “title”: “Connections Lab”,
  },
  { “code”: 202,
    “title”: “Critical Experiences”,
  }
}
```

http://api.nyu.edu/lowres/courses?title=connectionslab&year=2021

`http://api.nyu.edu/lowres/courses?title=connectionslab&year=2021`

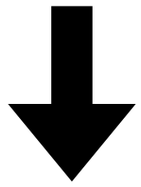
Query Parameters

http://api.nyu.edu/lowres/courses?title=connectionslab&year=2021



{ JSON DATA }

http://api.nyu.edu/lowres/courses?title=connectionslab&year=2021




```
{
 “code”: 201,
 “title”: “Connections Lab”,
 “credits”: 4
 “session”: “fall”,
 “day”: “Tuesday”
 “sections”: [
   0800,
   1100,
   2000
 ],
 “instructors”:[
    { “name”: “Mathura Govindarajan”,
      “contact”: “mathura.mg@nyu.edu"
    },
    { “name”: “Ruta Kruliauskaite”,
      “contact”: “ruta@nyu.edu“
    },
    { “name”: “Craig Protzel”,
      “contact”: “craig.protzel@nyu.edu",
    }
 ]
}
```

URL Only

URL + Key

URL + Key + Authentication

1) REQUEST - pokesearch.com

Our Server

2) RESPONSE - HTML, CSS, JS

Client

3) REQUEST - pokeapi.json

Pokemon Server

4**) RESPONSE** - JSON Data