

WEB SCRAPING

PAULO CÉSAR TUYA

ÍNDICE

- Módulo 1: Introducción a Web Scraping
- **Módulo 2: Herramientas para el Análisis de una Página Web**
 -
- Módulo 3: Web Scraping con Scrapy
- Módulo 4: Scraping en páginas dinámicas
- Módulo 5: Despliegue de un Spider

A decorative graphic on the left side of the slide, consisting of a vertical line of blue dots connected by a series of blue lines that branch out and zig-zag, resembling a circuit board or a stylized tree.

OBJETIVOS DE LA CLASE



CAPACIDADES

- Realizar peticiones HTTP
- Usar expresiones Regulares
- Entender y usar selectores CSS
- Entender la estructura de árbol de un documento HTML



HERRAMIENTAS

- Librería `requests`
- Librería `re`
- Librería `BeautifulSoup`



ANÁLISIS DE UNA PÁGINA WEB

PÁGINA EN NUESTRO NAVEGADOR

facebook

Correo electrónico o teléfono

Contraseña

Iniciar sesión

¿Olvidaste tu cuenta?

Facebook te ayuda a comunicarte y compartir con las personas que forman parte de tu vida.



Abre una cuenta

Es rápido y fácil.

Nombre

Apellido

Número de celular o correo electrónico

Contraseña nueva

Fecha de nacimiento

20

oct

1994

?

Sexo

☐ Mujer

☐ Hombre

☐ Personalizado

?

Al hacer clic en "Registrarte", aceptas nuestras Condiciones, la Política de datos y la Política de cookies. Es posible que te enviemos notificaciones por SMS, que puedes desactivar cuando quieras.

Registrarte

CÓDIGO HTML RECIBIDO POR EL NAVEGADOR

```
'<!DOCTYPE html>\n<html lang="en" id="facebook" class="no_js">\n<head><meta charset="utf-8" /><meta name="referrer" content="default" id="meta_referrer" /><script>window._cstart=new Date();</script><script>function envFlush(a){function b(b){for(var c in a)b[c]=a[c]}window.requireLazy?window.requireLazy(["Env"],b):(window.Env=window.Env||{}),b(window.Env)}envFlush({"ajaxpipe_token":"AXh3FFDpY8RBP_21","timeslice_hearthbeat_config":{"pollIntervalMs":33,"idleGapThresholdMs":60,"ignoredTimesliceNames":{"requestAnimationFrame":true,"Event listenHandler mousemove":true,"Event listenHandler mouseover":true,"Event listenHandler mouseout":true,"Event listenHandler scroll":true},"isHeartbeatEnabled":true,"isArtilleryOn":false},"shouldLogCounters":true,"timeslice_categories":{"react_render":true,"reflow":true},"sample_continuation_stacktraces":true,"dom_mutation_flag":true,"stack_trace_limit":30,"deferred_stack_trace_rate":1000,"timesliceBufferSize":5000,"show_invariant_decoder":false,"isQuick":false});</script><style></style><script>_DEV_=0;CavalryLogger=window.CavalryLogger||function(a){this.lid=a,this.transition=!1,this.metric_collected=!1,this.is_detailed_profiler=!1,this.instrumentation_started=!1,this.pagelet_metrics={},this.events={},this.ongoing_watch={},this.values={t_cstart:window._cstart},this.piggy_values={},this.bootloader_metrics={},this.resource_to_pagelet_mapping={},this.initializeInstrumentation&&this.initializeInstrumentation(),CavalryLogger.prototype.setIsDetailedProfiler=function(a){this.is_detailed_profiler=a;return this},CavalryLogger.prototype.setValue=function(a,b,c,d){d=d?this.piggy_values:this.values;(typeof d[a]===undefined||c)&&(d[a]=b);return this},CavalryLogger.prototype.getLastTtiValue=function(){return this.lastTtiValue},CavalryLogger.prototype.setTimestamp=CavalryLogger.prototype.setTimestamp||function(a,b,c,d){this.mark(a);var e=this.values.t_cstart|this.values.t_start;e=d?e+d:CavalryLogger.now();this.setValue(a,e,b,c);this.tti_event&&a===this.tti_event&&(this.lastTtiValue=e,this.setTimestamp("t_tti",b));return this},CavalryLogger.prototype.mark=typeof console===object&&console.timeStamp?function(a){console.timeStamp(a)}:function(){};CavalryLogger.prototype.addPiggyback=function(a,b){this.piggy_values[a]=b;return this},CavalryLogger.instances={},CavalryLogger.id=0,CavalryLogger.disableArtilleryOnUntilOffLogging=!1,CavalryLogger.getInstance=function(a){typeof a===undefined&&(a=CavalryLogger.id);CavalryLogger.instances[a]||(CavalryLogger.instances[a]=new CavalryLogger(a));return CavalryLogger.instances[a]},CavalryLogger.setPageID=function(a){if(CavalryLogger.id===0){var b=CavalryLogger.getInstance();CavalryLogger.instances[a]=b;CavalryLogger.instances[a].lid=a;delete CavalryLogger.instances[0]}CavalryLogger.id=a,CavalryLogger.now=function(){return window.performance&&performance.timing&&performance.timing.navigationStart&&performance.now?performance.now()+performance.timing.navigationStart:new Date().getTime()},CavalryLogger.prototype.measureResources=function(){},CavalryLogger.prototype.profileEarlyResources=function(){},CavalryLogger.getBootloaderMetricsFromAllLoggers=function(){};CavalryLogger.start_js=function(){};CavalryLogger.done_js=function(){};CavalryLogger.getInstance().setTTIEvent("t_domcontent");CavalryLogger.prototype.measureResources=function(a,b){if(!this.log_resources)return;var c="bootstrap/"+a.name;if(this.bootloader_metrics[c]!==void 0||this.ongoing_watch[c]!==void 0)return;var d=CavalryLogger.now();this.ongoing_watch[c]=d;"start_"+c in this.bootloader_metrics||(this.bootloader_metrics["start_"+c]=d);b&&!("tag_"+c in this.bootloader_metrics)&&(this.bootloader_metrics["tag_"+c]=b);if(a.type==="js"){c="js_exec/"+a.name;this.ongoing_watch[c]=d},CavalryLogger.prototype.stopWatch=function(a){if(this.ongoing_watch[a]){var b=CavalryLogger.now(),c=b-this.ongoing_watch[a];this.bootloader_metrics[a]=c;var d=this.piggy_values,a.indexOf("bootstrap")===0&&(d.t_resource_downloaded||(d.t_resource_downloaded=0),d.resources_downloaded||(d.resources_downloaded=0),d.t_resource_downloaded+=c,d.resources_downloaded+=1,d["tag_"+a]===_EF_&&(d.t_pagelet_cssload_early_resources=b));delete this.ongoing_watch[a];return this},CavalryLogger.getBootloaderMetricsFromAllLoggers=function(){var a={};Object.values(window.CavalryLogger.instances).forEach(function(b){b.bootloader_metrics&&Object.assign(a,b.bootloader_metrics)});return a},CavalryLogger.start_js=function(a){for(var b=0;b<a.length;++b)CavalryLogger.getInstance().stopWatch("js_exec/"+a[b]),CavalryLogger.done_js=function(a){for(var b=0;b<a.length;++b)CavalryLogger.getInstance().stopWatch("bootstrap/"+a[b]),CavalryLogger.prototype.profileEarlyResources=function(a){for(var b=0;b<a.length;b++)this.measureResources({name:a[b][0],type:a[b][1]?"js":"","_EF_"});CavalryLogger.getInstance().log_resources=true;CavalryLogger.getInstance().setIsDetailedProfiler(true);window.CavalryLogger&&CavalryLogger.getInstance().setTimestamp("t_start");</script><noscript><meta http-equiv="refresh" content="0";URL=?_fb_noscript=1" /></noscript><title id="pageTitle">Facebook - Log In or Sign Up</title><meta property="og:site_name" content="Facebook" /><meta property="og:url" content="https://www.facebook.com/" /><meta property="og:image" content="https://www.facebook.com/images/fb_icon_325x325.png" /><meta property="og:locale" content="en_US" /><meta property="og:locale:alternate" content="www" /><meta property="og:locale:alternate" content="es_LA" /></meta>
```




¿CÓMO ANALIZAR ESTE CÓDIGO?

- **Recibir** el código HTML de la página
- Entender la **estructura** del código
- Encontrar aquellas **secciones** que son de nuestro interés
- Buscar **patrones** en el contenido de la página



HERRAMIENTAS DE EXTRACCIÓN



¿CÓMO ANALIZAR UNA PÁGINA WEB?

- **Recibir** el código HTML de la página
 - Entender la **estructura** del código
 - Encontrar aquellas **secciones** que son de nuestro interés
 - Buscar **patrones** en el contenido de la página




RECIBIENDO EL CÓDIGO

- Necesitamos llevar el **código HTML** a Python
- Se debe realizar una solicitud mediante el protocolo **HTTP**
- Estándar de-facto en Python: librería `requests`




LIBRERÍA REQUESTS

- `pip install requests`
- Permite enviar peticiones HTTP/1.1  de manera sencilla e intuitiva
- Usa la tecnología de `urllib3` para permitir la conexión constante
- Posibilidades:
 - Hacer peticiones HTTP
 - Hacer queries con encabezado y mensaje
 - Inspeccionar la data de las respuestas
 - Hacer peticiones con autenticación
 - Configurar las peticiones para evitar sobrecargar al servidor



EJEMPLO DE USO DE REQUESTS

```
import requests
page = requests.get('https://github.com/trending')
if page.status_code is 200: 
    print(page.content)
```

```
b'\n\n\n\n\n\n\nrel="dns-pre
rel="dns-pre
rel="dns-pre
crossorigin=
href="https:
integrity="s
https://gith
integrity="s
https://gith
content="wid
people build
type="applic
title="Githu
property="og
content="Git
property="og
content="ima
content="htt
property="og
https://gith
property="og
property="tw
content="133
property="tw
million proj
property="tw
https://gith
name="select
content="KT5
name="google
name="octoly
name="octoly
name="octoly
name="octoly
class="js-ga
```

```
b'\n\n\n\n\n\n\nrel="dns-pre
rel="dns-pre
rel="dns-pre
crossorigin=
href="https:
integrity="s
https://gith
integrity="s
https://gith
content="wid
people build
type="applic
title="Githu
property="og
content="Git
property="og
content="ima
content="htt
property="og
https://gith
property="og
property="tw
content="133
property="tw
million proj
property="tw
https://gith
name="select
content="KT5
name="google
name="octoly
name="octoly
name="octoly
name="octoly
class="js-ga
```




LIBRERÍA REQUESTS

- <https://requests.kennethreitz.org/en/master/> - Documentación oficial de la librería requests
- <https://realpython.com/python-requests/> - Guía detallada del uso de la librería




¿CÓMO ANALIZAR UNA PÁGINA WEB?

- **Recibir** el código HTML de la página
- Entender la **estructura** del código
- Encontrar aquellas **secciones** que son de nuestro interés
- Buscar **patrones** en el contenido de la página

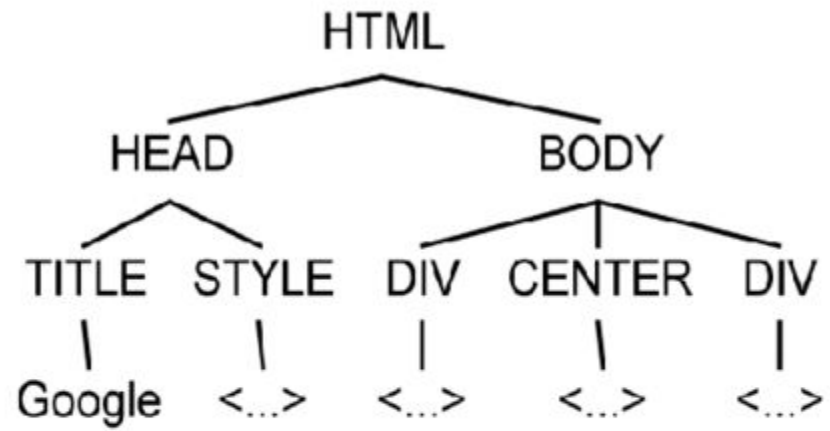


ESTRUCTURANDO EL CÓDIGO RECIBIDO

- El código HTML recibido por requests es idéntico al que recibe nuestro navegador
- Para poder trabajar con él necesitamos darle estructura
- Estándar de-facto en Python: BeautifulSoup 
- Es necesario entender la estructura de árbol del código HTML
- También es necesario familiarizarse con el contenido del código

CÓDIGO FORMATEADO VS ÁRBOL DE CÓDIGO HTML

```
<html>
  <head>
    <title>Google</title>
    <style>...</style>
  </head>
  <body>
    <div>...</div>
    <center>...</center>
    <div>...</div>
  </body>
</html>
```



INSPECCIONAR PÁGINA

Why GitHub? Enterprise Explore Marketplace Pricing Search GitHub Sign in Sign up

Explore Topics **Trending** Collections Events

Trending

See what the GitHub community is most excited about today.

Repositories Developers Language: Any Date range: Today

p.col-9.text-gray.my-1.pr-4 709.5 × 21 ★ Star

Tina is a site editing toolkit for modern React-based sites (Gatsby and Next.js)

TypeScript ★ 1,628 45 Built by [avatars] ★ 379 stars today

evilsocket / pwnagotchi ★ Star

(🐞 📡) - Deep Reinforcement Learning instrumenting bettercap for WiFi pwning.

Python ★ 1,757 201 Built by [avatars] ★ 90 stars today

ArduPilot / ardupilot ★ Star

ArduPlane, ArduCopter, ArduRover source

C++ ★ 4,406 8,404 Built by [avatars] ★ 6 stars today

ripenaar / free-for-dev ★ Star

Elements Console Sources Network Performance

```
<div class="Box-header d-md-flex flex-items-center flex-justify-between">
  </div>
  <div>
    <article class="Box-row">
      <div class="float-right"></div>
      <h1 class="h3 lh-condensed"></h1>
      <p class="col-9 text-gray my-1 pr-4"></p> == $0
      <div class="fe text-gray mt-2"></div>
    </article>
    <article class="Box-row"></article>
    <article class="Box-row"></article>
    <article class="Box-row"></article>
    <div class="float-right"></div>
    <h1 class="h3 lh-condensed">
      <a href="/ownthink/KnowledgeGraphData">
        <span class="mr-1 text-gray"></span>
        <span class="text-normal">ownthink /</span>
      </a>
    </h1>
  </div>
  KnowledgeGraphData
</div>
</h1>
```

html body div main div div.Box div article.Box-row p.col-9.text-gray.my-1.pr-4

Filter :hov .cls

element.style { }

```
.pr-4 {
  padding-right: 24px !important;
}

.my-1 {
  margin-top: 4px !important;
  margin-bottom: 4px !important;
}

.text-gray {
  color: #586069 !important;
}

.col-9 {
  width: 75%;
}

p {
  margin-top: 0;
  margin-bottom: 10px;
}
```

margin: 4px padding: 4px 24px 4px border: 1px solid #ccc

Filter Show all

- box-sizing
- border-box
- color
- rgb(88, 96, 105)
- display
- block
- font-family

Highlights from the Chrome 77 update



EJEMPLO DE USO DE BS

```
import requests
from bs4 import BeautifulSoup
page =
requests.get('https://github.com/trending')
if page.status_code is 200:
    content = BeautifulSoup(page.content,
    'html.parser')
    print(content)
```

CÓDIGO HTML ESTRUCTURADO

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="utf-8"/>
<link href="https://github.githubassets.com" rel="dns-prefetch"/>
<link href="https://avatars0.githubusercontent.com" rel="dns-prefetch"/>
<link href="https://avatars1.githubusercontent.com" rel="dns-prefetch"/>
<link href="https://avatars2.githubusercontent.com" rel="dns-prefetch"/>
<link href="https://avatars3.githubusercontent.com" rel="dns-prefetch"/>
<link href="https://github-cloud.s3.amazonaws.com" rel="dns-prefetch"/>
<link href="https://user-images.githubusercontent.com/" rel="dns-prefetch"/>
<link crossorigin="anonymous" href="https://github.githubassets.com/assets/frameworks-2fd1891c9e6292401a1a3de8bc3f747f.css" integrity="sha512-4bmhxCob3U2WoK8HV17UacoDdNejo+5081GN9SdGtjXbsQwP7" />
<link crossorigin="anonymous" href="https://github.githubassets.com/assets/site-09367dd1ae1784b858e71c8471ca0949.css" integrity="sha512-1N4+QlbnRp6tyqxbm0KgsAq2CenAm9g38Y5xUsD380+IUdydh071tbRr" />
<link crossorigin="anonymous" href="https://github.githubassets.com/assets/github-211e7a5168e3492fd5f3d4312f92593c.css" integrity="sha512-0+PsQv1dB+0GrYp3NVDj8G9tieBrfFhnagyspbrjFPXUyDi0Sv7Ffx" />
<meta content="width=device-width" name="viewport"/>
<title>Trending repositories on GitHub today · GitHub</title>
<meta content="GitHub is where people build software. More than 40 million people use GitHub to discover, fork, and contribute to over 100 million projects." name="description"/>
<link href="/opensearch.xml" rel="search" title="GitHub" type="application/opensearchdescription+xml"/>
<link href="https://github.com/fluidicon.png" rel="fluid-icon" title="GitHub"/>
<meta content="1401488693436528" property="fb:app_id"/>
<meta content="https://github.com" property="og:url"/>
<meta content="GitHub" property="og:site_name"/>
<meta content="Build software better, together" property="og:title"/>
<meta content="GitHub is where people build software. More than 40 million people use GitHub to discover, fork, and contribute to over 100 million projects." property="og:description"/>
<meta content="https://github.githubassets.com/images/modules/open_graph/github-logo.png" property="og:image"/>
<meta content="image/png" property="og:image:type"/>
<meta content="1200" property="og:image:width"/>
<meta content="1200" property="og:image:height"/>
<meta content="https://github.githubassets.com/images/modules/open_graph/github-mark.png" property="og:image"/>
<meta content="image/png" property="og:image:type"/>
<meta content="1200" property="og:image:width"/>
<meta content="620" property="og:image:height"/>
<meta content="https://github.githubassets.com/images/modules/open_graph/github-octocat.png" property="og:image"/>
<meta content="image/png" property="og:image:type"/>
<meta content="1200" property="og:image:width"/>
<meta content="620" property="og:image:height"/>
<meta content="github" property="twitter:site"/>
<meta content="13334762" property="twitter:site:id"/>
<meta content="github" property="twitter:creator"/>
<meta content="13334762" property="twitter:creator:id"/>
<meta content="summary_large_image" property="twitter:card"/>
<meta content="GitHub" property="twitter:title"/>
<meta content="GitHub is where people build software. More than 40 million people use GitHub to discover, fork, and contribute to over 100 million projects." property="twitter:description"/>
```




RECURSOS EN LÍNEA

- <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> - Documentación en línea de la librería BeautifulSoup




¿CÓMO ANALIZAR UNA PÁGINA WEB?

- **Recibir** el código HTML de la página
- Entender la **estructura** del código
- Encontrar aquellas **secciones** que son de nuestro interés
- Buscar **patrones** en el contenido de la página



ENCONTRAR LAS SECCIONES QUE NOS INTERESAN

- Sabemos que los archivos HTML tienen una estructura basada en etiquetas 
- Sin embargo, las etiquetas nos dicen poco y nada respecto a su contenido
- ¿Y si hubiese otro “nivel” en el cual podamos encontrar clasificaciones de secciones



HOJA DE ESTILO EN CASCADA (CSS)

- Las hojas de estilo en cascada permiten definir el estilo de un documento escrito en un lenguaje de marcado.
- La principal filosofía es separar contenido de formato.
- Permite aplicar distintas reglas de estilo basándose en clases
- Podemos utilizar estas clases para seccionar el contenido del documento



SELECTORES CSS

- Un selector CSS es una cadena de caracteres que describe qué secciones buscamos
- Los selectores CSS permiten hacer una **búsqueda avanzada** dentro de la estructura de un archivo HTML
- No necesitamos saber cómo se ve la información

SELECTORES MÁS COMUNES

Ejemplo	Selector	Descripción
*	Universal	Devuelve todos los elementos del documento
nombre	de Tipo	Devuelve los elementos con la etiqueta nombre
.nombre	de Clase	Devuelve los elementos con la clase nombre
#nombre	de ID	Devuelve los elementos con el ID nombre
[atrib]	de atributo	Devuelve los elementos que contengan el atributo atrib
[atrib="valor"]	de atributo	Devuelve los elementos cuyo valor de atrib sea valor



EJEMPLO DE USO DE SELECTORES CSS

```
import requests
```

```
from bs4 import BeautifulSoup
```

```
page = requests.get('https://github.com/trending')
```

```
if page.status_code is 200:
```

```
    content = BeautifulSoup(page.content, 'html.parser')
```

```
    boxrows = content.select('.Box-row')
```

```
    boxrows[0]
```


RESULTADO DE FILTRAR LA CLASE CSS BOX-ROW

```
<article class="Box-row">
<div class="float-right">
<a aria-label="You must be signed in to star a repository" class="btn btn-sm tooltipped tooltipped-s" data-hydro-click="{\"event_type\":\"authentication.click\",\"payload\":{\"location_in_page\":\"star button\",\"repository_id\":198488459,\"a
<svg aria-hidden="true" class="octicon octicon-star v-align-text-bottom" height="16" version="1.1" viewBox="0 0 14 16" width="14"><path d="M14 6l-4.9-.64L7 1 4.9 5.36 0 6l3.6 3.26L2.67 14 7 11.67 11.33 14l-.93-4.74L14 6z" fill-ru
Star
</a>
</div>
<h1 class="h3 lh-condensed">
<a href="/tinacms/tinacms">
<span class="mr-1 text-gray">
<svg aria-label="repo" class="octicon octicon-repo" height="16" role="img" version="1.1" viewBox="0 0 12 16" width="12"><path d="M4 9H3V8h1v1zm0-3H3v1h1V6zm0-2H3v1h1V4zm0-2H3v1h1V2zm8-1v12c0 .55-.45 1-1 1H6v2l-1.5-1.5L3 16v-2H1c-
</span>
<span class="text-normal">
tinacms /
</span>

tinacms
</a>
</h1>
<p class="col-9 text-gray my-1 pr-4">
Tina is a site editing toolkit for modern React-based sites (Gatsby and Next.js)
</p>
<div class="f6 text-gray mt-2">
<span class="d-inline-block ml-0 mr-3">
<span class="repo-language-color" style="background-color: #2b7489"></span>
<span itemprop="programmingLanguage">TypeScript</span>
</span>
<a class="muted-link d-inline-block mr-3" href="/tinacms/tinacms/stargazers.tinacms">
<span aria-label="star">
<svg aria-label="star" class="octicon octicon-star" height="16" role="img" version="1.1" viewBox="0 0 14 16" width="14"><path d="M14 6l-4.9-.64L7 1 4.9 5.36 0 6l3.6 3.26L2.67 14 7 11.67 11.33 14l-.93-4.74L14 6z" fill-rule="evenodd
</span>

1,626
</a>
<a class="muted-link d-inline-block mr-3" href="/tinacms/tinacms/network/members.tinacms">
<span aria-label="fork">
<svg aria-label="repo-forked" class="octicon octicon-repo-forked" height="16" role="img" version="1.1" viewBox="0 0 10 16" width="10"><path d="M8 1a1.993 1.993 0 00-1 3.72V6L5 8 3 6V4.72A1.993 1.993 0 002 1a1.993 1.993 0 00-1 3.7
</span>

45
</a>
<span class="d-inline-block mr-3">
Built by
```

SELECTOR TRAS SELECTOR

```
boxrows[0].select('.float-right')
```

```
[<div class="float-right">  
  <a aria-label="You must be signed in to star a repository" class="btn btn-sm tooltip tooltip-s" data-hydro-click='{ "event_  
  <svg aria-hidden="true" class="octicon octicon-star v-align-text-bottom" height="16" version="1.1" viewBox="0 0 14 16" width="14  
    Star  
  </a>  
</div>]
```



RECURSOS EN LÍNEA

- <https://www.w3schools.com/cssref/trysel.asp> - Permite probar tus selectores en base a un código HTML de ejemplo



¿CÓMO ANALIZAR UNA PÁGINA WEB?

- **Recibir** el código HTML de la página
- Entender la **estructura** del código
- Encontrar aquellas **secciones** que son de nuestro interés
- Buscar **patrones** en el contenido de la página




BUSCANDO PATRONES

- Un patrón puede entenderse como una **regularidad** en el mundo.
- El código estructurado **siempre** presentará diversos patrones.
- Incluso la información no estructurada **puede** presentar patrones.
- Para encontrar patrones en el texto utilizamos **expresiones regulares**.




EXPRESIONES REGULARES

- También conocidos como **RegEx**
- Secuencia de caracteres que forma un **patrón de búsqueda**.
- Permite identificar regularidades en **cadena de caracteres**.
- Se construye una **máscara** que busca todas las subcadenas que cumplan con ella
- Estructura de las expresiones regulares:
 - Caracteres: a, b, c, d, ...
 - Metacaracteres: \d, \w, \W, ... 
 - Operaciones: Alternación, Cuantificación, Agrupación



CARACTERES EN REGEX

- Todo símbolo que corresponda a un **grafema** del lenguaje escrito 
- **Cualquier símbolo** que pueda ser creado escribiendo con el teclado
- Es buscado por la máscara RegEx de manera **literal**
- Ejemplos:
 - a
 - b
 - c
 - d






METACARACTERES EN REGEX

- Caracteres que RegEx se reserva para poder utilizarlos con un **sentido especial**
- Permiten coincidir con **situaciones particulares** de la cadena o con **múltiples caracteres** a la vez
- Ejemplos:
 - **.** - Coincide con cualquier caracter excepto el fin de línea
 - **^** - Coincide con el fin de línea
 - **\$** - Coincide con el fin de la cadena

METACARACTERES EN REGEX

	Metacharacter	Metacharacter name	Meaning
1	^	caret	denote the beginning of a regular expression
2	\$	Dollar sign	denote the end of a regular expression or ending of a line
3	[]	Square bracket	check for any single character in the character set specified in []
4	()	Parenthesis	Check for a string. Create and store variables.
5	?	Question mark	check for zero or one occurrence of the preceding character
6	+	Plus sign	check for one or more occurrence of the preceding character
7	*	Multiply sign	check for any number of occurrences (including zero occurrences) of the preceding character.
8	.	Dot	check for a single character which is not the ending of a line
9		Pipe symbol	Logical OR
10	\	Escaping character	escape from the normal way a subsequent character is interpreted.
11	!	Exclamation symbol	Logical NOT
12	{}	Curly Brackets	Repeat preceding character

OPERACIONES EN REGEX

- Uso de **dos o más** caracteres o cadenas de caracteres para plasmar una situación única
- Utiliza **metacaracteres** como símbolos que representan las operaciones
- Tres tipos de operacion:
 - Alternación - `perro | gato | raton` 
 - Cuantificación - `ma{1,2}` 
 - Agrupación - `[0-9]` 

REGEX PARA DIRECCIONES DE CORREO

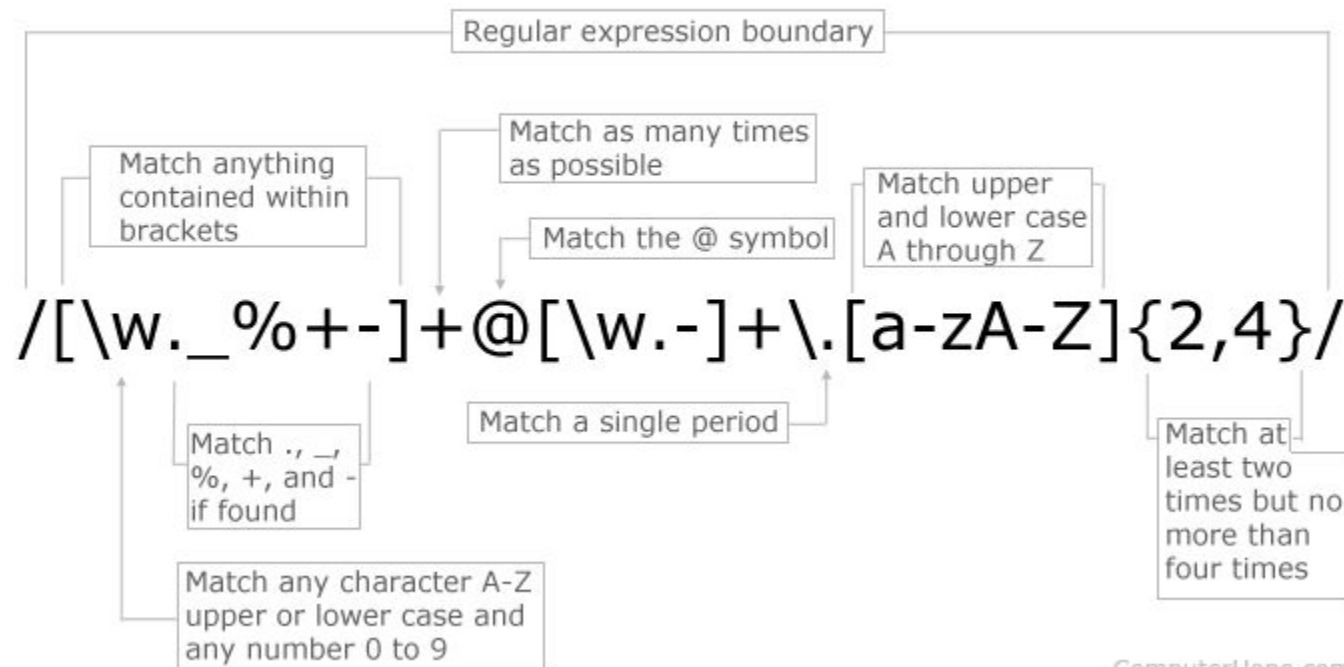
REGULAR EXPRESSION

```
!r" (^ [a-zA-Z0-9_ . + - ] + @ [a-zA-Z0-9 - ] + \. [a-zA-Z0-9 - . ] + $ )
```

TEST STRING

```
pctuya@pucp.pe  
pctuya@pucppe  
pctuya@pucppe.  
pctuyapucp.pe  
pctuya  
pctuya@pucp  
paulot@livemedia.pe
```

ESTRUCTURA DE UN REGEX



ComputerHope.com



RECURSOS EN LÍNEA

- <https://regexone.com/> - Curso intuitivo que les permitirá familiarizarse con el uso de RegEx
- <https://regex101.com/> - Herramienta online que les permitirá probar sus RegEx
- <https://docs.python.org/3/library/re.html> - Documentación de RegEx en Python



ACTIVIDAD MÓDULO 2

**GRACIAS
POR SU ATENCIÓN**