Guía P. O. O.

Programación Orientada a Objetos

1. ¿Cuál es el modelo para un objeto?
André: Es la clase a la que pertenece esa instancia (objeto).

2. Diferencia entre una clase y un objeto
Beto: El objeto instancia una clase

3. Que representa la clase en un objeto:
Beto: el modelo

4. Es el método que inicializa las propiedades de un objeto
El constructor

5. Cómo se accede a los elementos privados de una clase
Una vez encapsulada: con Getters y Setters

6. Los detalles internos de una clase pueden ser ocultados del exterior?
Si, Encapsulándolos

7. Dos clases base que heredan métodos y atributos a otra clase
Herencia Multiple (MALA PRÁCTICA)

Información coherente y relacionada con la clase:
Alta cohesión

La alta cohesión requiere:
Bajo acoplamiento

9. Qué es Polimorfismo:
Parámetros sobre un método y cambia su comportamiento

nota: sobrecarga:dependiendo de la cantidad y tipo de parámetros que te mandes es el método que va a trabajar.

10. Por que utilizar Polymer

encapsular los componentes de nuestra aplicación en etiquetas HTML5. Este futuro estará compuesto por tres tecnologías: Templates Shadow Dom, Custom Elements y HTML Imports

11. Como utilizamos la palabra Extends
class Hijo extends Padre, es utilizada una clase herede de otra

Qué palabra venía en el prefijo de la clase que hace referencia en Herencia
Extends

https://www.pensemosweb.com/

https://github.com/BBVA/1-ninjahack/wiki/Introducci%C3%B3n-a-Cells

https://github.com/pateroski/suricatos

Anexo
Herencia Jerárquica vs. Herencia Múltiple

Abstracción: llevar un fenómeno u objeto de la vida cotidiana a la lógica de programación

Beto: 5547759343

Guía JavaScript

1. Tipos de Datos

Tipos primitivos:
undefined
Nulo (null)
Símbolo (symbol) [similar a una bandera] NO
Lógicos (boolean)
Numérico (number)
Cadena (string)
De tipo Object : (con constructor)
Object Array, Date y Promise

R: Number, object, string, boolean, null, undefined

2. Diferencia Let, Var, CONST

VAR: una variable declarada fuera de una función es una variable global y es pasada por referencia a scopes descendientes o herederos
LET: Si la variable es declarada let dentro un bloque que a su vez está dentro de una función, la variable pertenece solo a ese bloque
CONST: directamente prohíbe la reasignación de valores y solo permite el valor asignado directamente.

*¿cuando ocupar let y var?
Cuando queremos limitar el alcance de la variable

3. ¿COMO SE CONVIERTE UN JSON A STRING?
JSON.parse()

4.¿Como se convierte un Objeto de javascript a JSON?
JSON.stringify()

5.¿Que es un Array?
Es un objeto global que es usado en la construcción de arreglos

6.¿Nombres de eventos Javascript?

evento	Descripcion	Elementos para los que está definido
onblur	Cuando el foco se pierde	<button>, <input>, <label>, <select>, <textarea>, <body>
onchange	Cuando existe un cambio en el elemento	<input>, <select>, <textarea>
onclick	Cuando se clickea una vez en un elemento	Todos los elementos
ondblclick	Cuando se clickea dos vez en un elemento	Todos los elementos
onfocus	Cuando el elemento gana foco	<button>, <input>, <label>, <select>, <textarea>, <body>
onkeydown	Cuando se pulsa una tecla y no se suelta	Elementos de formulario y <body>
onkeypress	Cuando se presiona una tecla	Elementos de formulario y <body>
onkeyup	Cuando se deja de presionar una tecla	Elementos de formulario y <body>
onload	El momento en el que se carga el DOM	<body>
onmousedown	pulsar el botón del ratón y no soltarlo	Todos los elementos
onmousemove	Cuando el mouse cambia de posición	Todos los elementos
onmouseout	Cuando el cursor deja de estar en el elemento	Todos los elementos
onmouseover	Cuando el mouse se posiciona sobre un elemento	Todos los elementos
onmouseup	Soltar el botón del ratón	Todos los elementos
onreset	inicializar el formulario	<form>
onresize	Cuando el elemento cambia de dimensiones	<body>
onselect	Cuando un elemento se selecciona	<input>, <textarea>

evento	Descripcion	Elementos para los que está definido
onsubmit	Cuando hay un envío de información	<form>
onunload	Se abandona la página, por ejemplo al cerrar el navegador	<body>

7.-COMO PASAR UN OBJETO A JAVASCRIPT?
JSON.parse()

8.-COMO Y CUANDO UTILIZAR UNA VARIABLE VAR Y LET
VAR: como variable de tipo global y en todo el documento js
LET: como variable dentro de un bloque de código o dentro de una estructura de control

R=Cuando queremos limitar el alcance de la variable

9. ¿Qué es una Promesa?
Es un objeto que representa la terminación o el fracaso eventual de una operación asíncrona.

10.- ¿Qué es un Arreglo?

Son objetos tipo lista de alto nivel.

11.- ¿En qué consiste la estructura de control “for”?

Crea un bucle que consiste en tres expresiones opcionales

```
for(let a = 0 ; a< arr.length ; a++){

}
```

12.- Describe el funcionamiento de un “forEach”

Ejecuta la función indicada una vez por cada elemento del array.

13.- If, else

```
If(condición que retorne un boolean){
        acciones a realizar si se cumple la condición
} else {
        acciones a realizar si no se cumple la condición
}
```

14.- Custom Events

```
// agrega Listener
element.addEventListener('mi-evento', () => { metodo(e.detail) })

// crea evento personalizado
const MyEvent = new CustomEvent('mi-evento', {details: tipo de dato});

// disparar el evento
elemento.dispatchEvent(MyEvent)
```

```
let letras = ["lambda", "alfa", "gamma", "beta"];
letras.sort();
```

Que hace Sort ?

Ordena alfabéticamente el contenido de un arreglo

Resultado de este código document.querySelector('p.five')

R= selecciona el primer elemento párrafo con una clase five

Qué resultado es de esta función:

```
function myFuncion(){
const nombre = "pedro"
nombre = "juan"
return nombre
}
```

```
function foo(){
return 5
}
let myVar = foo;
```

R= la variable myVar hace referencia a la función

Que retorna la siguiente función

```
function f (){
var x; var y = x === null;
 console.log(y)
} f();
```

false

¿Que es lo que hace el siguiente código?

```
if(x===y){
console.log(“hola mundo”);
}
```

Compara que “x” y “y” sean exactamente iguales tanto en tipo y valor

Cual de los sig estados es una promesa
R= Rechazado, aceptado y en espera

document.querySelector("p.five");
todos los elemento <p> con clase "five"
el primer elemento <p> con clase "five"
primeros 5 elementos p

```
let mascotas = ["perro", "gato", "loro", "canario"];
mascotas.filter(mascota=>mascota.length >=5 );
console.log(mascotas);
```

[“loro”,”canario”]

Correo:

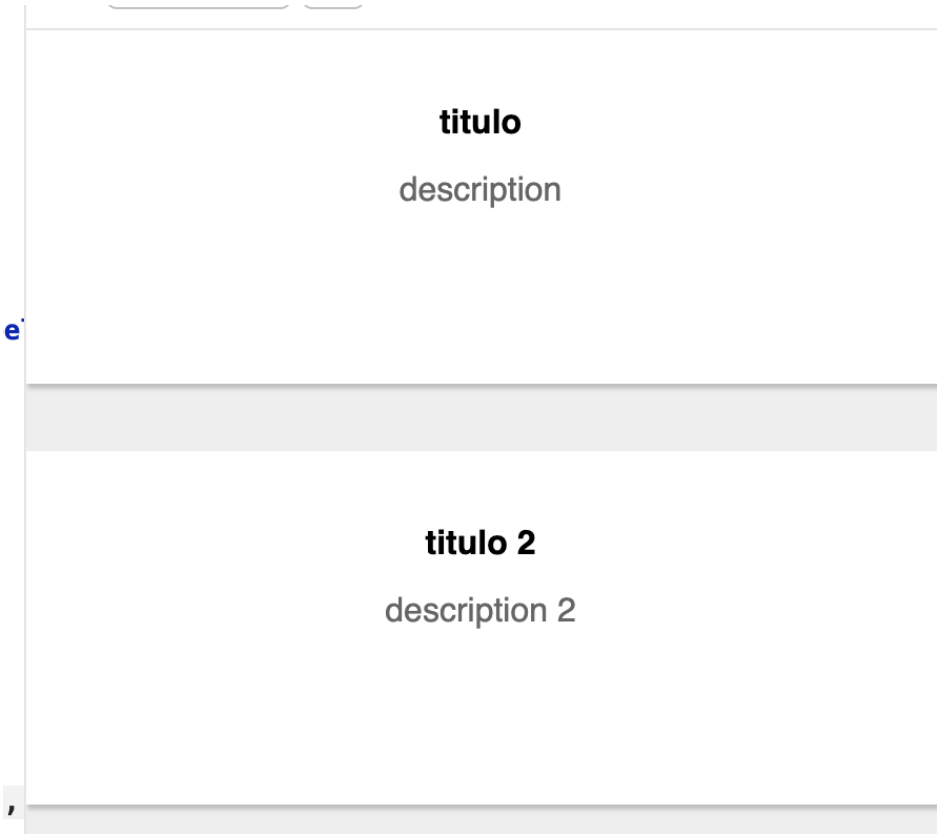
aserranoh@ids.com.mx

Para que vayan midiendo sus Skills en los diversos codigos:
https://medium.com/coderbyte/the-10-best-coding-challenge-websites-for-2018-12b57645b654

Ejercicio polymer fin de semana

En el siguiente ejercicio se necesita encontrar y corregir los errores dentro del código y obtener como resultado final, como se muestra en la imagen

https://glitch.com/edit/#!/picayune-aphid?path=index.html:20:20



Ejercicios Polymer 07/06/2019

Keys no válidas en en una declaración polymer:
<R= readyToVerify, notify, readOnly, type

Observers de qué forma son:
R= sincrona

Como se pone un listenner a un campo de texto, para saber en que momento se pierde el focus:
< R= on-blur

Como se realiza el monitoreo de cambios en las propiedades de un elemento
< R = observer, databinding, computed

Como se llama un observer de dos propiedades:
< R: observer complejo

Propiedades virtuales cuyo valor se determina basado en otras propiedades
< R: computed

Cuales son los tipos de datos válidos en polymer
< R = number, string, array, object, date, boolean

Como eliminar el valor de un arreglo en Polymer
R: con el metodo splice, pop, shift
<R R = this.splice('persons', 2, 1) (este es el que vino en nuestro examen)

El tipo de dato que representa un nulo :
<R = Object

De forma declarativa como se hace binding en un atributo de un componente
R: [[]] y {{}} (si puede ser de one-way or me)
< R= <first-name R= <mi-componente first-name="{{}}" >

Como hacer un binding dentro de un elemento input
R: {{"nombre::input"}}
< R= {{firsName::input}}

Que tipo de bindeo es este: {{}}
R: two-way data binding

Cómo declarar un dom-if, dom-repeat

<template is="dom-repeat" items="[[propPadre]]" as="algo"
</template>

<template is="dom-if" if="[[boolean]]"
</template>

Que propiedad modifica su valor dependiendo de otros atributos
R: Computada (Computed)

Sintaxis correcta de un component

<mi-componente> </mi-componente>

Properties mas utilizados
<R = type, reflectToAtributte, value

Que es un custom event y como se dispara
R: Es un evento personalizado y se dispara con un dispatchEvent

De un arreglo de personas (objetos) modificar el 5to elemento
R= this.set('person.4.lastName', 'nuevo valor')

Dentro de un componente como cambiar el color

```
R = ::host {
var(--var-host-color , blue)
}
```

Permite uso de la función filter como una de sus propiedades
R: dom-repeat

Ciclos de vida de un componente de polymer:
R= Constructor, ConnectCallback, DisconnectCallback, attributeChangedCallback,

Es el propio constructor de la clase. El constructor se llama una vez por cada instancia que es invocada.
* connectedCallback() & disconnectedCallback()

Estos estados del ciclo de vida ocurren cuando los componentes se inyectan y borran del DOM de la página,

* attributeChangedCallback()

Se produce cada vez que un atributo o propiedad del componente cambia.

* ready()
Básicamente es como un "connectedCallback", solo que no se ejecuta más de una vez.

Que evento se dispara la siguiente línea de código?

```
name: {
    type: String,
    notify: true
}
this.set('name',  '');
```

Posibles respuestas:
R= name-changed

Cual de estos estados, rechaza una promesa
received ( )
rejected (X)
declined ( )
rejected ( )

Notifica a polymer del cambio hecho solamente en una propiedad

a) this.push ( )
b) this.notify ()
c) this.notifyPath (X)
d) this.changeProperty ( )

```
<template is="dom-repeat" items="{{persons}}" as="person">
  <template is="dom-if" if=[[!_graterThan(person.age,18)]] >
    <p>[[person.name]] [[person.lastName]] => [[person.age]]</p>
    </template>
  </template>
```

POO :

Qué palabra venía en el prefijo de la clase que hace referencia en Herencia
R= Extends

¿Cual de estas oraciones definen herencia?
R= La clase empleado tiene subclases Empleado Sindicalizado, Empleado…

¿Cual es el modelo para un objeto?
R= Clase

Diferencia entre una clase y un objeto
R=El objeto instancia una clase

Que representa la clase en un objeto:
R= modelo

Es el método que inicializa las propiedades de un objeto
R= El constructor

Cómo se accede a los elementos privados de una clase
R=Encapsulamiento

Los detalles internos de una clase pueden ser ocultados del exterior
R=Encapsulamiento

Dos clases base que heredan métodos y atributos a otra clase
R= Herencia Múltiple

Super clase con dos hijos
R= Herencia

Información coherente y relacionada con la clase:
R: alta cohesión

La alta cohesión con que debe de ir:
R= Bajo acoplamiento

Cual de las siguientes define una herencia
R= un padre empleados pasa sus atributos a empleado del mes etc

Qué es Polimorfismo:
R= Parámetros sobre un método y cambia su comportamiento

JavaScript

Que tipo de dato representa un null
R= Objeto

Menciona las variables de Javascript
R= var let const

Menciona los tipos de variables en Javascript:
R=Number, object, string, boolean, null, undefine

¿Como pasar un String JSON a un Objeto Javascript?
R=JSON.parse()

Como pasar un objeto javascript a String JSON?
R=JSON.stringify()

¿Cómo y cuándo utilizar las variables let y var?
R=Cuando queremos limitar el alcance de la variable

Afirmaciones, cuál no es cierta en Javascript:
R = poner ; al final de una oración

```
let letras = ["lambda", "alfa", "gamma", "beta"];
letras.sort();
```

R = ordena el arreglo de forma ascendente

Resultado de este código document.querySelector('p.five')
R= selecciona el primer elemento párrafo con una clase five

Elimina el elemento avión del siguiente arreglo:

```
let transportes = ["avion","barco","motocicletas","automovil"];
transportes.shift();
console.log(transportes);
```

R = transportes.shift();

Qué resultado es de esta función:

```
function myFuncion(){
const nombre = “pedro”
nombre = “juan”
return nombre
}
```

R = Uncaught TypeError: Assignment to constant variable.

```
function f(){
  let x= "4"+4+5;
  let y= 4+4+"5";
  console.log(x + ' '+ y );
  console.log(type of);
   }
f();
```

R = Javascript hace type casting para convertir los tipos a strings, “445” “85”

Que resultado da:

```
var square = number => number*number;
```

R= te da un número al cuadrado una funcion con dos parametros

```
// Que retorna la siguiente function
  function f(x, y=2, z=7){
  return x + y + z;
  }
  console.log(f(1) === 10);
```

R = true

```
function foo(){
return 5
}
let myVar = foo;
```

R= la variable myVar hace referencia a la función

Como agregar un valor al final del arreglo

```
arr = ["val1","val2", "val3"];
console.log(arr.length  )
R=          arr[arr.length ] = "value"
console.log(arr);
```

```
function f (){
var x; var y = x === null;
 console.log(y)
} f();
```

R = false

```
function f (){
var x; var y = x === null;
 console.log(y)
} f();
```

R = false

```
(function f(f){
return typeof f();
})(function(){return 1;})
```

R= regresa number

¿Que es lo que hace el siguiente código?

```
if(x===y){
console.log(“hola mundo”);
}
```

R= Hace la comparación de tipo y valor e imprime “hola mundo”

Cual de los sig estados es una promesa

R= Rechazado, aceptado y en espera

document.querySelector("p.five");
todos los elemento <p> con clase "five"

el primer elemento <p> con clase "five"
primeros 5 elementos p

```
let mascotas = ["perro", "gato", "loro", "canario"];
mascotas.filter(mascota=>mascota.length >=5 );
console.log(mascotas);
```