|  |  |  |
| --- | --- | --- |
|  | IT Department  Final exam**: November 23th 2018**  Web Applications Development  TC2026 | |
| Professor: Eduardo Aguirre Bermúdez | Grade: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Name: Natanael Pérez Bucio | | Student ID: A01371010 |

**Instructions: Read carefully each question and choose the most accurate answer where applicable. You will also find some open questions where you must write the answer by yourself. You are not allowed to use any kind of material.**

**Each question has a value of 3.6 points. Therefore the total value of the exam is 100.8 points.**

1. Provide the code to link a web page to a CSS file ***mystyles.css.***

<link href=”mystyles.css” rel=”stylesheet”>

1. What are inline styles, embedded styles, and external style sheets?

Inline styles are the styles that are given for each element in specific

Emmbedded styles are the ones that are linked by the css stylesheet

External styles are the ones that are contained in other css stylesheets that are also linked to the HTML, but not the main css

1. What keyword do you add to a style property to override style precedence and style inheritance?

!important

1. Provide a selector to match all ***address*** elements that are direct children of the ***footer*** element.

footer > address {

}

1. The initial ***h1*** heading in a document has the ID ***top***. Provide a style rule to display the text of this ***h1*** heading in ***Century Gothic, Helvetica***, or a ***sans-serif*** font.

.top {

font-family: Century Gothic, Helvetica or a sans-serif

}

1. For the following style rules, what is the font size of the ***h1*** heading in pixels?

body {font-size: 16px;}

body > article {font-size: 0.75em;}

body > article > h1 {font-size: 1.5em;}

18px

1. Provide the style rule to set the ***padding*** around every ***h1*** in a ***section*** element to ***1em*** on ***top***, ***0.5em*** on the ***left*** and ***right***, and ***2em*** on the ***bottom***.

section h1 {

padding-top: 1em;

padding-left: 0.5em;

padding-right: 0.5em;

padding-bottom: 2em;

}

1. Describe the items selected by the following selector:

div.Links img[usemap]

all the images that has usemap as an attribute. Also this images has to be into a div that has the Links class

1. Provide the style rule to display all ***hypertext*** ***links*** within a ***navigation list*** as ***block*** elements with a ***gray*** ***background***.

nav a {

display: block;

background-color: gray;

}

1. Provide a style rule to display the ***footer*** element only after all ***floated elements*** have cleared.

footer {

clear: both;

}

1. Your layout has four floated elements in a row but unfortunately the last element has wrapped to a new line. What is the source of the layout mistake?

The sizes of the layouts had exceeded the maximum width that it’s admissible for the row, then the layout deviates the last element to a new row.

1. What is the difference between relative positioning and absolute positioning?

The absolute positioning has a stated point for it’s displaying while the relative positioning depends on the other elements for be positioned.

1. One element has a ***z-index*** value of ***1***; a second element has a ***z-index*** value of ***5***. Will the second element always be displayed on top of the first? Explain why or why not.

The element with the z-index of 1 will be always on top, because the greatest the index is more it’s priority to be displayed.

1. How should you arrange the media queries in your style sheet if you want to support mobile, tablet, and desktop devices?

Desktop, tablet and mobile

1. Provide a style to display ***flexbox*** items in a single line, oriented ***vertically***.

.element {

display: flex;

flex-direction: column;

}

1. Provide a style that sets the ***growth rate*** of the ***flex*** item to ***4***.

.element {

display: flex;

}

.element-child {

Flex-grow: 4;

}

1. What are the two phases in the lifecycle of a client-side web application?

The phase for starting the webpage and the phase for listening events

1. How many events can be processed at once?

One

1. In the following code snippet, which functions are callback functions?

numbers.sort(function **sortAsc**(a,b){

return a – b;

});

function **ninja**(){}

ninja();

var myButton = document.getElementById("myButton");

myButton.addEventListener("click", function **handleClick**(){

alert("Clicked");

});

1. Within the body of the test function, what are the values of parameters a, b, and c for the two function calls?

function test(a, b, ...c){ /\*a, b, c\*/}

test(1, 2, 3, 4, 5);

test();

first: a = 1 b = 2 c = [ 3, 4, 5 ]

second: a = undefined b = undefined c = [ ]

1. After executing the following code snippet, what are the values of the ***message1*** and ***message2*** variables?

function getNinjaWieldingWeapon(ninja, weapon = "katana"){

return ninja + " " + weapon;

}

var message1 = getNinjaWieldingWeapon("Yoshi");

var message2 = getNinjaWieldingWeapon("Yoshi", "wakizashi");

message1: Yoshi katana

message2: Yoshi wakizashi

1. After running the following code, what are the values of variables **ninja** and **samurai**?

function getSamurai(samurai){

"use strict"

arguments[0] = "Ishida";

return samurai;

}

function getNinja(ninja){

arguments[0] = "Fuma";

return ninja;

}

var samurai = getSamurai("Toyotomi");

var ninja = getNinja("Yoshi");

ninja: {

“Yoshi”:

[“Fuma”]

}

samurai: {

“Toyotomi”

}

1. When running the following code, which of the assertions will pass? ***(circle them)***

function whoAmI1(){

"use strict";

return this;

}

function whoAmI2(){

return this;

}

assert(whoAmI1() === window, "Window?");

assert(whoAmI2() === window, "Window?");

1. When running the following code, which of the assertions will pass? ***(circle them)***

var ninja1 = {

whoAmI: function(){

return this;

}

};

var ninja2 = {

whoAmI: ninja1.whoAmI

};

var identify = ninja2.whoAmI;

assert(ninja1.whoAmI() === ninja1, "ninja1?");

assert(ninja2.whoAmI() === ninja1, " ninja1 again?");

assert(identify() === ninja1, "ninja1 again?");

assert(ninja1.whoAmI.call(ninja2) === ninja2, "ninja2 here?");

1. When running the following code, which of the assertions will pass? ***(circle them)***

function Ninja(){

this.whoAmI = () => this;

}

var ninja1 = new Ninja();

var ninja2 = {

whoAmI: ninja1.whoAmI

};

assert(ninja1.whoAmI() === ninja1, "ninja1 here?");

assert(ninja2.whoAmI() === ninja2, "ninja2 here?");

1. Which of the following assertions will pass? ***(circle them)***

function Ninja(){

this.whoAmI = function(){

return this;

}.bind(this);

}

var ninja1 = new Ninja();

var ninja2 = {

whoAmI: ninja1.whoAmI

};

assert(ninja1.whoAmI() === ninja1, "ninja1 here?");

assert(ninja2.whoAmI() === ninja2, "ninja2 here?");

1. What’s the output of the following code?

const promise = new Promise((resolve, reject) => {

reject("Hattori");

});

promise.then(val => alert("Success: " + val))

.catch(e => alert("Error: " + e));

“Error: Hattori”

1. What’s the output of the following code?

const promise = new Promise((resolve, reject) => {

resolve("Hattori");

setTimeout(()=> reject("Yoshi"), 500);

});

promise.then(val => alert("Success: " + val))

.catch(e => alert("Error: " + e));

“Success: Hattori”