

# Question #1 (HTML5 / CSS3)

What size will "Text" be?

```
<style type="text/css">
    body { font-size: 24pt; }
    p { font-size: 50%; }
    span { font-size: 1.5em; }
</style>

<p><span>Text<span></p>
```

1.  18pt

2.  1.5pt

3.  12pt

4.  24pt

5.  36pt

## Question #2 (HTML5 / CSS3)

Which of the following code samples are valid?

1.

```
<p class="abc">
  <div>
    Lorem ipsum dolor sit amet, <small>consectetuer adipiscing</small> elit,
    sed diem nonummy nibh <b>euismod tincidunt</b> ut <span>lacreer dolore
    magna aliquam</span> erat volutpat.
  </div>
</p>
```

2.

```
<div class="abc">
  <p>
    Lorem ipsum <i>dolor</i> sit amet, <small>consectetuer adipiscing</small>
    elit, sed diem nonummy nibh <b>euismod tincidunt</b> ut
    <span>lacreer dolore magna aliquam</span> erat volutpat.
  </p>
</div>
```

3.

```
<p class="abc">
  <ul>
    <li>Text1</li>
    <li>Text2</li>
  </ul>
</p>
```

4.

```
<div class="abc">
  <p>
    Lorem ipsum <i>dolor</i> sit amet, <small>consectetuer adipiscing</small>
    elit, sed diem nonummy nibh <b>euismod tincidunt</b> ut
    <p>lacreer dolore magna aliquam</p>erat volutpat.
  </p>
</div>
```

## Question #3 (HTML5 / CSS3)

Which of the following will create a grid with 4 columns of equal width?



```
.grid-equal-columns {  
    display: grid;  
    /* your answer */  
}
```

1.  `grid-columns: 25% 4;`
2.  `grid-template-columns: repeat(4, 1fr);`
3.  `columns: 25% 25% 25% 25%;`
4.  `grid-columns: 1 1 1 1;`
5.  `columns: 4;`
6.  `grid-template-columns: auto;`

## Question #4 (HTML5 / CSS3)

Which of the following sets of elements can be used to provide a caption to an image?

1.

```
<picture>
  <img ... />
  <alt>Caption Text</alt>
</picture>
```

2.

```
<object>
  <img ... />
  <description>Caption Text</description>
</object>
```

3.

```
<photo>
  <img ... />
  <text>Caption Text</text>
</photo>
```

4.

```
<figure>
  <img ... />
  <figcaption>Caption Text</figcaption>
</figure>
```

## Question #5 (HTML5 / CSS3)

Which of the following is the valid method for providing multiple-resolution images to different devices?

1.  ``
2.  `<picture src="640w.jpg"  
mobile="320w.jpg"  
tablet="480w.jpg"  
desktop="640w.jpg">`
3.  ``
4.  ``

## Question #6 (HTML5 / CSS3)

We're going to create a new CSS Variable with the value "1rem".  
Which of these is the correct syntax?

```
:root {  
    /* your answer */  
}
```

1.  \$padding: 1rem;
2.  --padding: 1rem;
3.  padding: var(1rem);
4.  var(--padding): 1rem;

## Question #7 (HTML5 / CSS3)

Which selector adds the style to the first row of the table?

```
<table>
  <tr><td></td></tr>
  <tr><td></td></tr>
</table>
```

- table tr:first
- table tr:first-letter
- table tr:first-row
- table tr:first-child
- None of the above

# Question #8 (HTML5 / CSS3)

Which of the following is the equal style to

```
span { margin: 10px 5px 15px; }
```

1.

```
span { margin: 10px 5px 15px 10px; }
```

2.

```
span { margin: 10px 5px 15px 15px; }
```

3.

```
span { margin: 10px 5px 15px 5px; }
```

4.  The style is invalid.

## Question #9 (HTML5 / CSS3)

Assuming the following CSS, and using the default CSS Box Model, which statement is true:

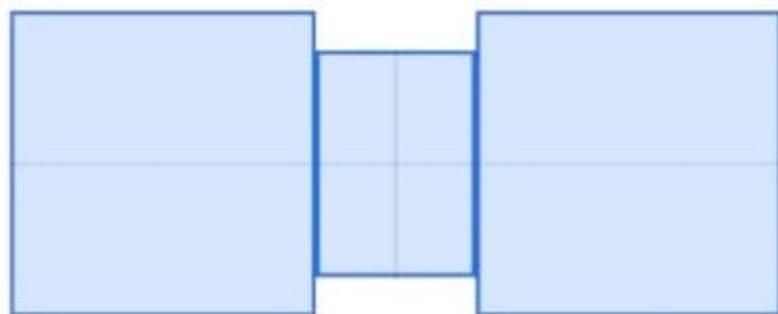
```
.box {  
    width: 400px;  
    padding: 10px;  
    border: 1px;  
    margin: 20px;  
}
```

- The width defines the size of the content area.  
Padding, border, and margin are then added to that width to make the total size the item takes up on screen.  
Therefore, the total width of this box will be **462px**.
- The width defines the total width of the area used by the element, including margins, padding, and borders.  
The width of this box will be **400px**.
- The width defines the content area plus padding and border, but not margins.  
The width of the box will therefore be **422px**.

## Question #10 (HTML5 / CSS3)

The flex container shown below has three children elements.

Which **two** properties, when applied to a flex container, will center its flex items along both axes?



1.  vertical-align: middle;

2.  justify-self: center;

3.  justify-content: center;

4.  text-align: center;

5.  align-items: center;

6.  align-self: center;

## Question #11 (HTML5 / CSS3)

We have the following HTML/CSS code:

```
<style>
  p { color: red; }

  @media (max-width:400px) {
    p { color: blue; }
  }

  @media (min-width:800px) {
    p { color: green; }
  }

  @media (min-width:200px) and (max-width:600px) {
    p { color: black; }
  }
</style>

<p>Some text</p>
```

When the text will be red?

1.  When the screen width is below **200px**.
2.  When the screen width is above **800px**.
3.  When the screen width is between **600px** and **800px**.
4.  When the screen width is below **400px** or above **600px**.
5.  Never.

# Question #12 (HTML5 / CSS3)

Consider the following HTML code:

```
<style type="text/css">
    #first { border: 1px solid black; }
    .second { border-color: red; }
</style>

<div id="first" class="second">Text 1</div>
```

What will the border color of the <div> element be?

1.  Black
2.  Red
3.  The <div> element will not have a border.

## Question #13 (HTML5 / CSS3)

Data attributes can be used to add specific data to an element on the page. Which of the following will style the element with the value "win"?

```
<li data-result="win">3-0 to Tom Richards</li>
```

1.

```
[result="win"] {  
    background: green;  
}
```

2.

```
data-result:win {  
    background: green;  
}
```

3.

```
:data(result="win") {  
    background: green;  
}
```

4.

```
[data-result="win"] {  
    background: green;  
}
```

## Question #14 (HTML5 / CSS3)

Given the following markup, which selector would give all direct children (the "p" and "ul" elements) a red border, but not the grandchildren (the "li" elements)?

```
<div class="container">
  <p>Paragraph 1</p>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
</div>
```

- .container p, ul
- .container p ul
- .container > \*
- .container + \*

## Question #15 (HTML5 / CSS3)

If I want images to scale down when they are larger than their containing block, but not to be stretched larger, I would:

1.  Set **max-width: 100%** on the image.
2.  Set **width: 0%** on the image.
3.  Set **width: 100%** on the image.
4.  Set **width: contain** on the image.

## Question #16 (HTML5 / CSS3)

Consider the following HTML code:

```
<style type="text/css">
    .first { margin: 10px 0px; }
    .second { margin: 20px 0px; }
</style>

<div class="first">Text 1</div>
<div class="second">Text 2</div>
```

What will the distance between "Text 1" and "Text 2" be?

1.  10px
2.  0px
3.  30px
4.  20px

## Question #17 (HTML5 / CSS3)

We have the following HTML/CSS code:

```
<style>
  .container {
    display: flex;
    border: 5px solid;
  }

  .box {
    height: 100px;
    width: 100px;
    background: red;
  }
</style>
<div class="container">
  <div class="box"></div>
  <div class="box"></div>
</div>
```

What CSS declaration should we add to the `.container` in order to obtain the following layout:



1.  `justify-content: space-between;`
2.  `justify-content: space-inside;`
3.  `align-items: space-inside;`
4.  `align-items: space-between;`

## **Question #18** (HTML5 / CSS3)

In which cases will the style for the selector "div + p { color: red; }" be applied?

1.  The style will be applied to <p> tag placed after <div> tag.
2.  The style will be applied to <div> tag containing <p> tag.
3.  The style will be applied to <div> tag placed before <p> tag.
4.  The style will be applied to <p> tag placed inside <div> tag.
5.  The selector is invalid.

# Question #19 (HTML5 / CSS3)

What color will be applied to the text?

```
<style>
  span.text {color:red;}
  .text span {color:blue}
  div {color:green}
  div span {color:black}
  p span {color:yellow}
</style>

<div>
  <p>
    <span class="text"><span>Text</span></span>
  </p>
</div>
```

1.  Red
2.  Blue
3.  Green
4.  Black
5.  Yellow

## Question #20 (HTML5 / CSS3)

Given this fragment of HTML code:

```
<link type="text/css" href="style.css" />
<style type="text/css">
    div { height: 10px!important; }
</style>
<div style="height: 15px;"></div>
```

And this style.css:

```
div { height: 20px; }
```

What will the height of the <div> element be?

- 20px
- 30px
- 10px
- 15px

## Question #21 (JavaScript)

What values will the following JavaScript code log to the console?

```
var counter = (function() {
    var myCounter= 0;

    function changeBy(val) {
        myCounter += val;
    }

    return {
        increment: function () {
            changeBy(1);
        },
        decrement: function () {
            changeBy(-1);
        },
        value: function () {
            return myCounter;
        }
    };
})();

console.log(counter.value());
counter.increment();
counter.increment();
console.log(counter.value());
counter.decrement();
console.log(counter.value());
counter.changeBy(2);
console.log(counter.value());
```

1.  0, 2, 1
2.  0, 2, 1, 3
3.  0, 0, 0, 2
4.  0, 0, 0

## Question #22 (JavaScript)

What values will the following JavaScript code output to the console?

```
var x = 1;

function foo() {
    var x = 3;
}
foo();
console.log(x);

function bar() {
    x = 4;
}
bar();
console.log(x);
```

1.  3, 4
2.  4, 4
3.  1, 4
4.  3, 3
5.  1, 1

## Question #23 (JavaScript)

What is a “scoping function” useful for?

1.  Keeping some data and functions private while revealing others
2.  Avoiding creating global variables and functions
3.  Creating a shared, private scope for the code and data inside it
4.  All of the above

## Question #24 (JavaScript)

Which of the following would be the correct way to read the "age" property of a "person" object?

1.

`person::age`

2.

`person[age]`

3.

`person.age`

4.

`person['age']`

## Question #25 (JavaScript)

Which of the following are valid ways to create an object in JavaScript?

1.

```
var person = { FirstName: "John", LastName: "Galt" };
```

2.

```
var person = { "FirstName": "John", "LastName": "Galt" };
```

3.

```
var person = { FirstName = "John", LastName = "Galt" };
```

4.

```
var person = new { FirstName = "John", LastName = "Galt" };
```

5.

```
var person = new Object();

person.FirstName = "John";

person.LastName = "Galt";
```

6.

```
var person = new Object(FirstName = "John", LastName = "Galt");
```

## Question #26 (JavaScript)

With the following HTML and JavaScript, the alert isn't shown when you click the button. Why not?

```
<input type="button" value="Click Me" onclick="handleClick">
<script>
  (function setup() {
    function handleClick() {
      alert("Clicked!");
    }
  })();
</script>
```

- There should be () after handleClick in the onclick attribute
- handleClick should be a global function
- The setup function runs before the DOM is ready
- The setup function never runs

## Question #27 (JavaScript)

What is the output of this JavaScript?

```
function f(x) {
    x += 1;
}

function g(x) {
    x.value *= 5;
}

var a;
var b = 1;
var c = { value: 2 };
var d = c;

f(a);
f(b);
g(d);

console.log(a, b, c.value, d.value);
```

1.  undefined 1 2 2

2.  NaN 2 2 10

3.  1 2 2 10

4.  undefined 1 10 10

## **Question #28** (JavaScript)

Which of the following techniques can be used to increase performance of JavaScript code?

1.  Avoid using the with keyword
2.  Reduce activity in loops
3.  Reduce Dom access
4.  Reduce Dom size

## Question #29 (JavaScript)

The following code is intended to add five identical boxes containing links to the document, but it doesn't work properly. Why not?

```
// Copies the contents of one box into another
function copyContents(from, to) {
    for (var i = 0; i < from.childNodes.length; i++) {
        to.appendChild(from.childNodes[i]);
    }
}

// Create a box to copy
var referenceBox = document.createElement('div');

var link = document.createElement('a');
link.href = 'https://www.example.com/';
link.textContent = 'A link';

referenceBox.appendChild(link);

// Add box copies to the document
for (var i = 0; i < 5; i++) {
    var newBox = document.createElement('div');
    copyContents(referenceBox, newBox);

    document.body.appendChild(newBox);
}
```

- to.appendChild() expects HTML, but from.childNodes[i] is a node object, so all the boxes will contain the text [Object Node].
- document.createElement() reuses existing elements with the same tag, so only one box is added to the document.
- A link's href has to be set using setAttribute(); setting the property link.href won't do anything, so none of the links in the boxes will point anywhere.
- The same link element can't have multiple parents, so only one box ends up with a link in it.

# Question #30 (JavaScript)

What will this code output?

```
console.log(typeof a);
console.log(typeof b);

function a() {
}

var b = function() {
};
```

1.  function, undefined
2.  undefined, undefined
3.  undefined, function
4.  function, function
5.  An error

## Question #31 (JavaScript)

Which of the following can be used to handle the user clicking on a node?

1.  `node.attachEvent("onclick", myFunction)`
2.  `node.onclick(myFunction)`
3.  `node.addEventListener("click", myFunction, false)`
4.  `node.addEventListener("onclick", myFunction, false)`

## Question #32 (JavaScript)

After the following JavaScript code is run which of the following are true?

```
function foo(a, b, c) {  
    a++;  
    b = "new string";  
    c["key"] = "new value";  
}  
  
var a = 1,  
    b = "old string",  
    c = {"key" : "old value"};  
  
foo(a, b, c);
```

1.  **a** is 1
2.  **a** is 2
3.  **b** is "old string"
4.  **b** is "new string"
5.  **c** is {"key" : "old value"}
6.  **c** is {"key" : "new value"}

## Question #33 (JavaScript)

Given that this HTML and JavaScript

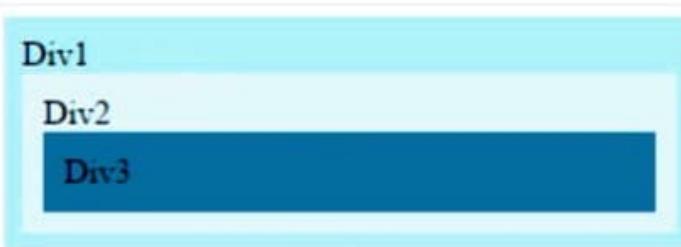
```
<html lang="en">
  <body>
    <div id='1' style='background-color:#bce8f1; padding: 10px'>
      Div1
      <div id='2' style='background-color:#d9edf7; padding: 10px'>
        Div2
        <div id='3' style='background-color:#3a87ad; padding: 10px'>
          Div3
        </div>
      </div>
    </div>
  </body>
</html>
```

```
var divs = document.getElementsByTagName('div');

for (var i = 0; i < divs.length; i++) {
  divs[i].addEventListener("click", logEvent, false);
}

function logEvent() {
  console.log(this.id);
}
```

will produce the following:



Div1  
Div2  
Div3

What values will be logged to the console when a user clicks on Div2?

1.  2
2.  2, 1
3.  2, 3
4.  None of the above.

## Question #34 (JavaScript)

Assuming the function produces elements on the page based on input from an untrusted source – for example, a query string ?input=... – which of these functions **do not** create a potential XSS vulnerability?

1. 

```
function innerHTMLStripped(input) {
    var element = document.createElement('div');
    var b = document.createElement('b');

    b.innerHTML = input;

    for (var i = 0; i < b.childNodes.length; i++) {
        if (b.childNodes[i].nodeType !== Node.TEXT_NODE) {
            b.removeChild(b.childNodes[i]);
        }
    }

    element.appendChild(b);

    return element;
}
```

2. 

```
function createTextNode(input) {
    var element = document.createElement('div');
    var b = document.createElement('b');

    b.appendChild(document.createTextNode(input));
    element.appendChild(b);

    return element;
}
```

3. 

```
function textContent(input) {
    var element = document.createElement('div');
    var b = document.createElement('b');

    b.textContent = input;
    element.appendChild(b);

    return element;
}
```

4. 

```
function innerHTMLEscaped(input) {
    var escaped = input
        .replace(/&/g, '&amp;')
        .replace(/</g, '&lt;')
        .replace(/>/g, '&gt;');

    var element = document.createElement('div');
    element.innerHTML = '<b>' + escaped + '</b>';

    return element;
}
```

5. 

```
function innerHTML(input) {
    var element = document.createElement('div');
    element.innerHTML = '<b>' + input + '</b>';

    return element;
}
```

## **Question #35** (JavaScript)

Which of the following statements are true?

1.  JavaScript can create cookies.
2.  JavaScript can react to events.
3.  JavaScript can read and write HTML elements.

## Question #36 (JavaScript)

What are the values of **x** and **y** after the following JavaScript code is run?

```
var x = 2;
var y = 4;

if ((y > x || y++ === 4) && ++y === 5) {
    x = 1;
} else {
    x = 4;
}
```

1.  x = 4, y = 4

2.  x = 1, y = 6

3.  x = 4, y = 5

4.  x = 1, y = 5

## Question #37 (JavaScript)

Which of the following would be the correct way to define and invoke a method that will display the "FirstName" property of a "person" object?

1.

```
person.ShowName = new function() { alert(FirstName); }  
person.ShowName();
```

2.

```
person.ShowName = Function.Create({alert(FirstName)});  
person.ShowName();
```

3.

```
person.ShowName = function(sender) { alert(sender.FirstName); }  
person.ShowName();
```

4.

```
person.ShowName = function() { alert(sender.FirstName); }  
person.ShowName();
```

5.

```
person.ShowName = function () { alert(this.FirstName); }  
person.ShowName();
```

6.  You cannot define a method for the object in JavaScript.

## Question #38 (JavaScript)

After the following JavaScript code is run which of the following is true?

```
var a = ['apple', 'banana', 'coconut']
var b = a
var c = a.slice()
a.push('date')
```

- b = ['apple', 'banana', 'coconut', 'date'] and c = ['apple', 'banana', 'coconut']
- b = ['apple', 'banana', 'coconut'] and c = ['apple', 'banana', 'coconut']
- b = ['apple', 'banana', 'coconut'] and c = ['apple', 'banana', 'coconut', 'date']
- b = ['apple', 'banana', 'coconut', 'date'] and c = ['apple', 'banana', 'coconut', 'date']

# Question #39 (JavaScript)

Which of the following JavaScript conditions are true?

1.

```
'1' == 1
```

2.

```
'1' === 1
```

3.

```
false == 0
```

4.

```
false === 0
```

5.

```
false == ''
```

6.

```
false === ''
```

7.

```
'abc' == 'abc'
```

8.

```
[1, 2, 3] == [1, 2, 3]
```

## Question #40 (JavaScript)

After the following JavaScript code is run which of the following statements will evaluate to true?

```
var Person = function(firstName, lastName, dateOfBirth, measurements){  
    this.firstName = firstName;  
    this.lastName = lastName;  
    this.dateOfBirth = dateOfBirth;  
    this.measurements = measurements  
}  
  
person = new Person('Jim', 'Perice', new Date(2013, 2, 12), {weight: '70kg'});  
personClone = JSON.parse(JSON.stringify(person));
```

1.  person.measurements['weight'] === personClone.measurements['weight']
  
2.  person.dateOfBirth.toDateString() === personClone.dateOfBirth.toDateString()
  
3.  person.firstName === personClone.firstName
  
4.  person === personClone