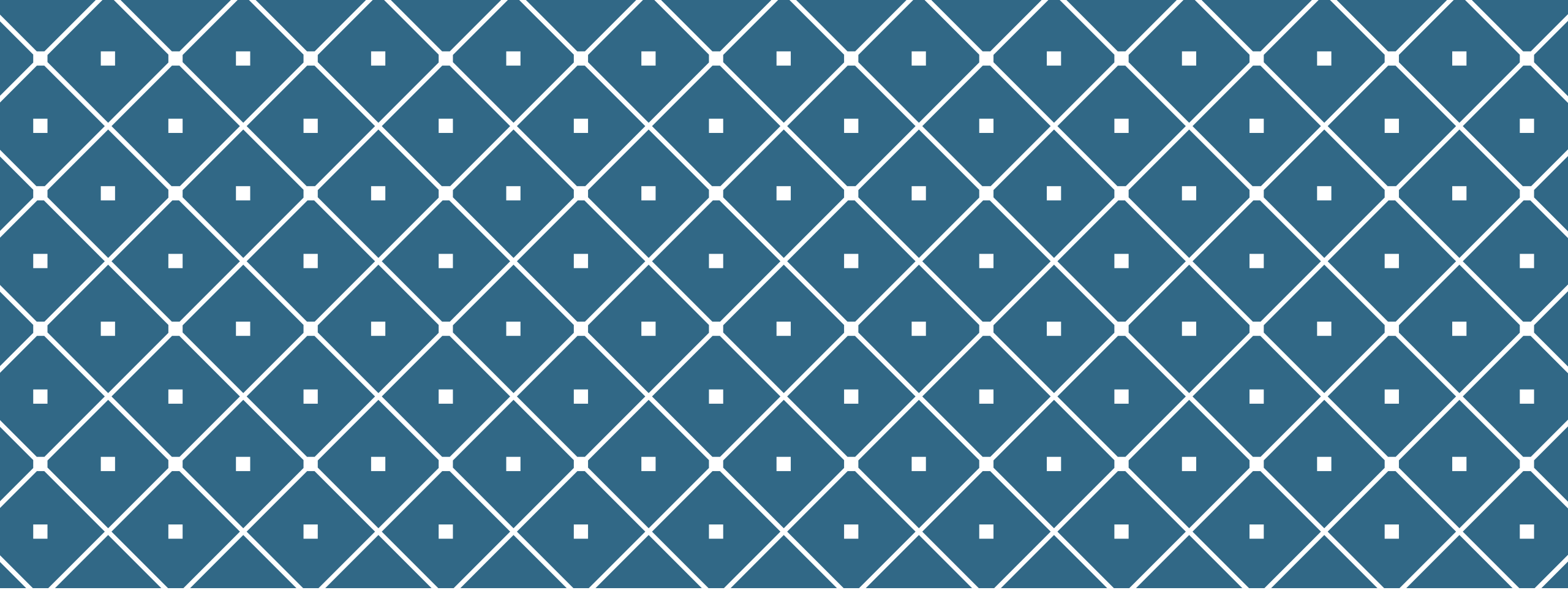
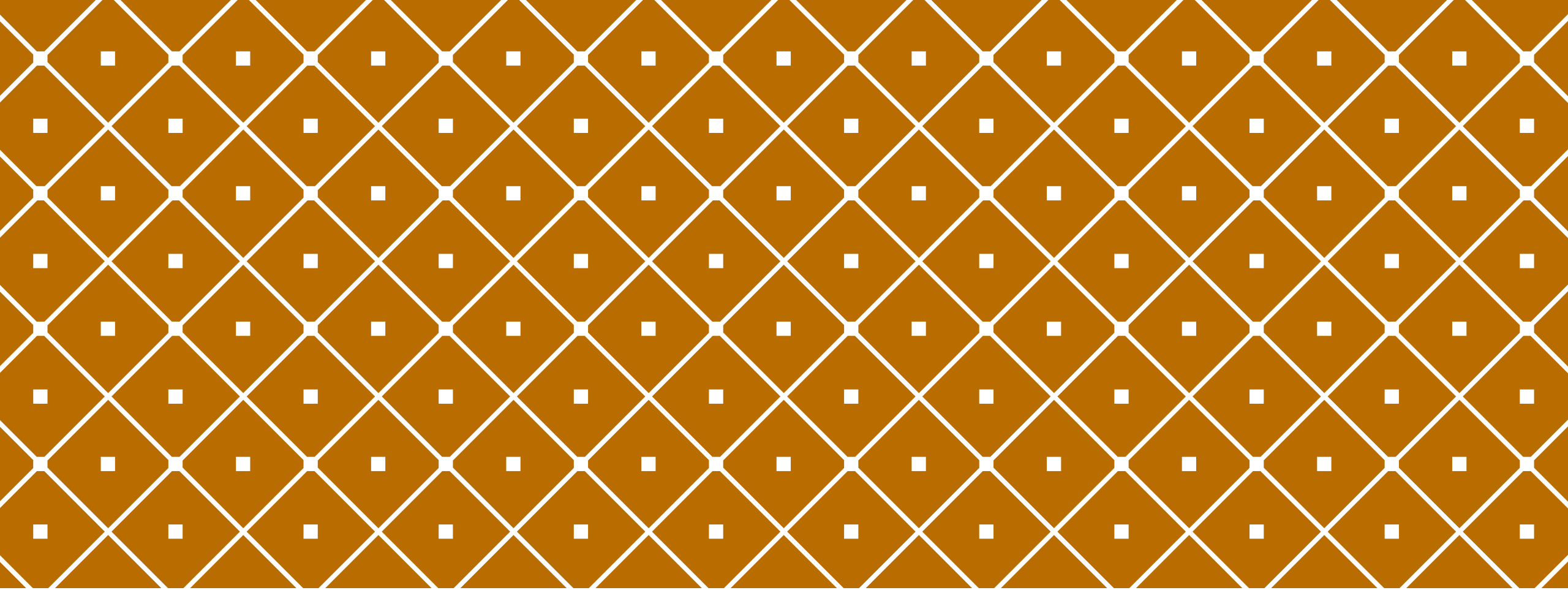


COMPUTER PROGRAMMING 2

John Paul C. Navarro



JAVASCRIPT PROGRAMMING

WHAT IS JAVASCRIPT?

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java.

JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

WHY LEARN JAVASCRIPT?

JavaScript is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

WHY LEARN JAVASCRIPT?

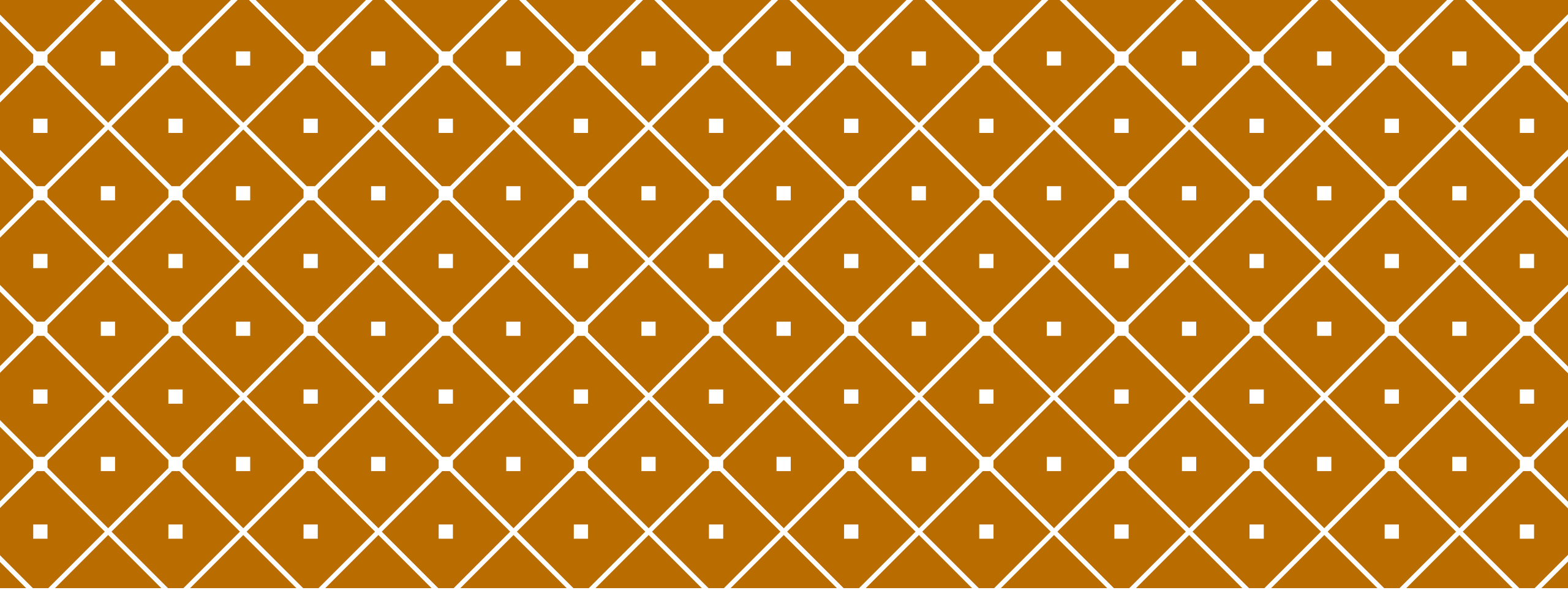
1. JavaScript is the most popular programming language in the world and that makes it a programmer's great choice.
2. JavaScript is everywhere, it comes installed on every modern web browser and so to learn JavaScript you really do not need any special environment setup.
3. JavaScript helps you create really beautiful and crazy fast websites.
4. JavaScript usage has now extended to mobile app development, desktop app development, and game development.

HELLO WORLD USING JAVASCRIPT

```
<html>
   <body>
      <script language = "javascript" type = "text/javascript">
         <!--
            document.write("Hello World!")
         //-->
      </script>
   </body>
</html>
```

JAVASCRIPT FRAMEWORKS

1. Angular
2. React
3. jQuery
4. Vue.js
5. Ext.js
6. Ember.js
7. Meteor
8. Mithril
9. Node.js
10. Polymer
11. Aurelia
12. Backbone.js



JAVASCRIPT - SYNTAX

JAVASCRIPT - SYNTAX

- JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page.
- You can place the <script> tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the <head> tags.
- The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

JAVASCRIPT - SYNTAX

The script tag takes two important attributes –

- **Language** – This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- **Type** – This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

JAVASCRIPT - SYNTAX

So your JavaScript segment will look like −

```
<script language = "javascript" type = "text/javascript">
   JavaScript code
</script>
```

WHITESPACE AND LINE BREAKS

JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs.

You can use spaces, tabs, and newlines freely in your program and you are free to format and indent your programs in a neat and consistent way that makes the code easy to read and understand.

SEMICOLONS ARE OPTIONAL

Simple statements in JavaScript are generally followed by a semicolon character, just as they are in C, C++, and Java.

JavaScript, however, allows you to omit this semicolon if each of your statements are placed on a separate line.

SEMICOLONS ARE OPTIONAL

```
<script language = "javascript" type = "text/javascript">
   <!--
      var1 = 10
      var2 = 20
   //-->
</script>
```

SEMICOLONS ARE OPTIONAL

But when formatted in a single line as follows, you must use semicolons –

```
<script language = "javascript" type = "text/javascript">
   <!--
      var1 = 10; var2 = 20;
   //-->
</script>
```

Note – It is a good programming practice to use semicolons.

CASE SENSITIVITY

JavaScript is a case-sensitive language. This means that the language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

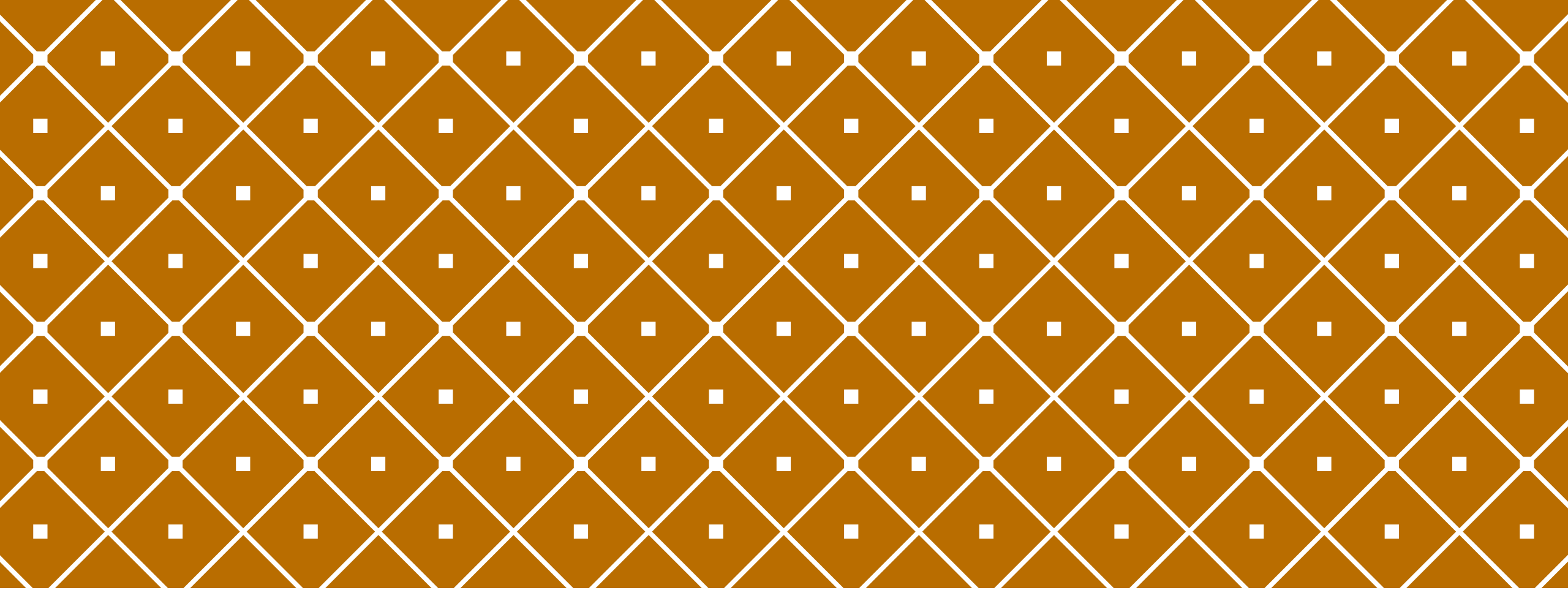
So the identifiers Time and TIME will convey different meanings in JavaScript.

NOTE – Care should be taken while writing variable and function names in JavaScript.

COMMENTS IN JAVASCRIPT

```
<script language = "javascript" type = "text/javascript">
   <!--
      // This is a comment. It is similar to comments in C++

      /*
      * This is a multi-line comment in JavaScript
      * It is very similar to comments in C Programming
      */
   //-->
</script>
```



JAVASCRIPT - PLACEMENT IN HTML FILE

JAVASCRIPT PLACEMENT

There is a flexibility given to include JavaScript code anywhere in an HTML document. However the most preferred ways to include JavaScript in an HTML file are as follows −

- Script in <head>...</head> section.
- Script in <body>...</body> section.
- Script in <body>...</body> and <head>...</head> sections.
- Script in an external file and then include in <head>...</head> section.

JAVASCRIPT IN <HEAD>...</HEAD> SECTION

If you want to have a script run on some event, such as when a user clicks somewhere, then you will place that script in the head as follows −

JAVASCRIPT IN <HEAD>…</HEAD> SECTION

```
<html>
<head>
    <script type="text/javascript">
        <!--
        function sayHello() {
            alert("Hello World")
        }
        //-->
    </script>
</head>
<body>
    <input type="button" onclick="sayHello()" value="Say Hello" />
</body>
</html>
```

JAVASCRIPT IN <BODY>...</BODY> SECTION

If you need a script to run as the page loads so that the script generates content in the page, then the script goes in the <body> portion of the document.

In this case, you would not have any function defined using JavaScript.

JAVASCRIPT IN <BODY>...</BODY> SECTION

```
<html>
    <head>
    </head>

    <body>
        <script type = "text/javascript">
            <!--
                document.write("Hello World")
            //-->
        </script>

        <p>This is web page body </p>
    </body>
</html>
```

JAVASCRIPT IN EXTERNAL FILE

As you begin to work more extensively with JavaScript, you will be likely to find that there are cases where you are reusing identical JavaScript code on multiple pages of a site.

You are not restricted to be maintaining identical code in multiple HTML files. The script tag provides a mechanism to allow you to store JavaScript in an external file and then include it into your HTML files.

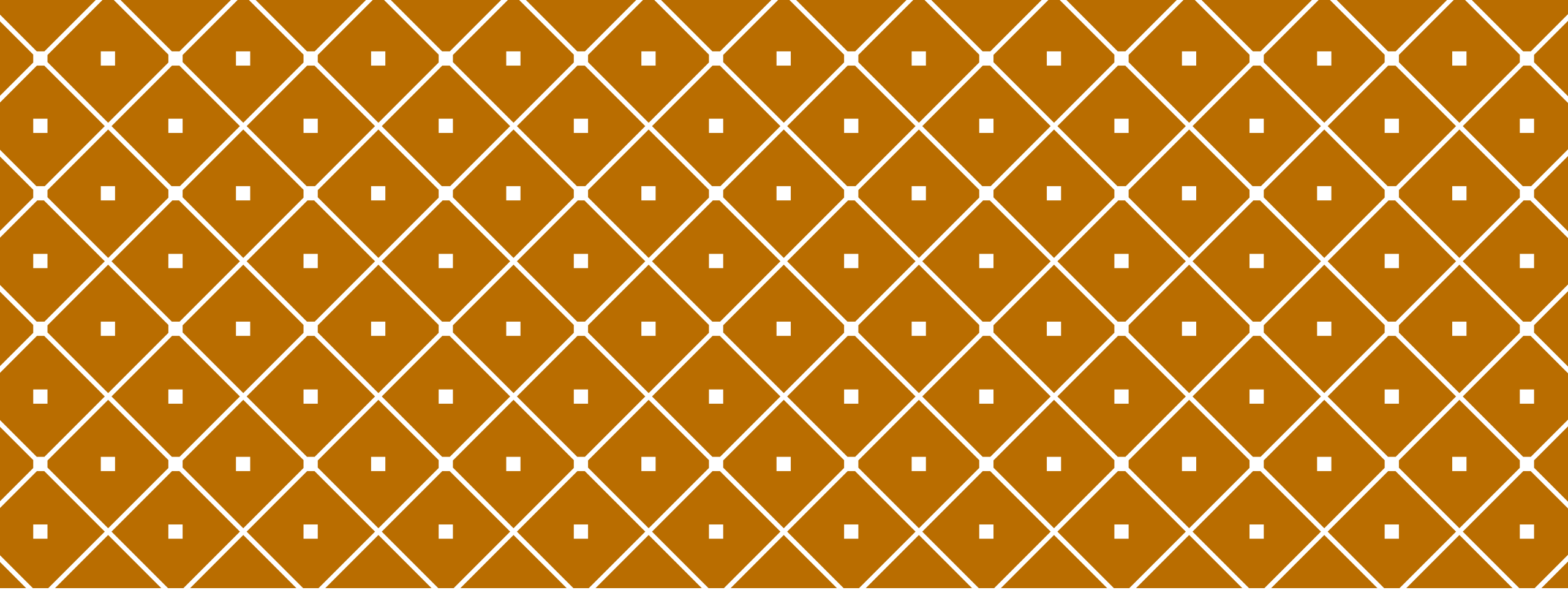
JAVASCRIPT IN EXTERNAL FILE

```
<html>
   <head>
      <script type = "text/javascript" src = "filename.js" ></script>
   </head>

   <body>
      .......
   </body>
</html>
```

JAVASCRIPT IN EXTERNAL FILE

```
function sayHello() {
    alert("Hello World")
}
```



JAVASCRIPT - VARIABLES

JAVASCRIPT DATATYPES

One of the most fundamental characteristics of a programming language is the set of data types it supports. These are the type of values that can be represented and manipulated in a programming language.

JavaScript allows you to work with three primitive data types –

- **Numbers,** eg. 123, 120.50 etc.
- **Strings** of text e.g. "This text string" etc.
- **Boolean** e.g. true or false.

JAVASCRIPT VARIABLES

Like many other programming languages, JavaScript has variables. Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container.

Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the var keyword as follows.

JAVASCRIPT VARIABLES

```
<script type = "text/javascript">
   <!--
      var money;
      var name;
   //-->
</script>
```

JAVASCRIPT VARIABLE SCOPE

The scope of a variable is the region of your program in which it is defined. JavaScript variables have only two scopes.

- **Global Variables** – A global variable has global scope which means it can be defined anywhere in your JavaScript code.
- **Local Variables** – A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

JAVASCRIPT VARIABLE SCOPE

```
<html>
   <body onload = checkscope();>
      <script type = "text/javascript">
         <!--
            var myVar = "global";      // Declare a global variable
            function checkscope( ) {
               var myVar = "local";    // Declare a local variable
               document.write(myVar);
            }
         //-->
      </script>
   </body>
</html>
```

JAVASCRIPT VARIABLE NAMES

- You should not use any of the JavaScript reserved keywords as a variable name. These keywords are mentioned in the next section. For example, **break** or **boolean** variable names are not valid.
- JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore character. For example, **123test** is an invalid variable name but **_123test** is a valid one.
- JavaScript variable names are case-sensitive. For example, **Name** and **name** are two different variables.

LET'S TRY THIS!

```
<!DOCTYPE html>
<html>
<body>

<h2>Enter your name:</h2>

<input type="text" id="nameInput" value="">
<button onclick="showName()">Show Name</button>

</body>
</html>
```

LET'S TRY THIS!

```
function showName() {
  var name = document.getElementById("nameInput").value;
  alert("Hello, " + name + "!");
}
```

CHALLENGE!

Create a JavaScript code that will do the following:

1. Ask the user for details: Name, Age, and E-mail.
2. Display the details provided by the user.
3. Upload here ➔ Sec 1: bit.ly/1AComProg2

   Sec 2: bit.ly/1BComProg2