***<!—Q1 -->***

*//A*

let length = 16 ;

console.log("typeof (A): " + typeof length);

*// B*

let name = ("Jack" )

console.log("typeof (B): " + typeof name);

*// C*

let number = '120' ;

console.log("typeof (C): " + typeof number);

*// D*

var y = 'undefined' ;

console.log("typeof (D): " + typeof y );

*// E*

let z = undefined ;

console.log("typeof (E): " + typeof z );

*// F*

let isAdmin = true ;

console.log("typeof (F): " + typeof isAdmin );

*// G*

const array = [1, 2, 'john' , true, '201', undefined] ;

console.log("typeof (G): " + typeof array );

*// H*

const x = {

firstName : 'ali' ,

lastName : 'hosseini'

} ;

console.log("typeof (H): " + typeof x );

***<!—Q2-->***

*// Q2 For number*

let input = +prompt("pls Enter your input : ");

alert(" type Of input : " + typeof(input));

*// Q2 for string*

let input = prompt("pls Enter your input : ");

alert(" type Of input : " + typeof(input));

***<!—Q3 -->***

let userName = prompt("Enter your name:");

let userName\_check = confirm("Please check your name '" + userName + "' ?  ");

*if* (userName\_check) {

    alert( " Hello '" + userName + "'");

}

*else* {

    alert(" No ! Please try again.");

}

***<!—Q4 -->***

let numberOne = +prompt(" Please Enter The Number one : ");

let numberTwo = +prompt(" Please Enter The Number two : ");

alert(" Result :" + ( numberOne + numberTwo) );

***<!-- Q5 -->***

<a *href*="https://www.google.com/" *target*="\_blank"> Go To The <em>Google</em> </a>

***<!—Q6 -->***

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | undefined | null | نوع |
| یک مقدار عددی صحیح است | نشان‌دهنده عدم تعیین مقدار یک متغیر است . | نشان‌دهنده عدم وجود یا تهی بودن مقدار است . | تعریف |
| let number = 0; | let myNumber;  console.log(myNumber);  // Output: undefined | let myNumber = null; | مثال |

***<!—Q7 -->***

|  |  |  |
| --- | --- | --- |
| فرق let , const با var | | |
| متغیر var | متغیر let | ثابت const |
| می‌توان به آن مقدار اولیه نداد. | می‌توان به آن مقدار اولیه نداد. | حتما باید مقدار اولیه داشته باشد. |
| قابلیت مقداردهی دوباره (تغییر مقدار) دارد. | قابلیت مقداردهی دوباره (تغییر مقدار) دارد. | قابلیت مقداردهی دوباره (تغییر مقدار) ندارد. |
| قابلیت تعریف ). مجدد دارد (متغیرهای var می‌توانند دوباره با var تعریف شوند | قابلیت تعریف مجدد ندارد. | قابلیت تعریف مجدد ندارد. |
| متغیر var یک Global Object می‌سازد و رم بیشتری نسبت به let و const مصرف می‌کند. | متغیر let یک Global Object نمی‌سازد. | متغیر const یک Global Object نمی‌سازد. |
| متغیر var، قابلیت Hoisting دارد که باعث می‌شود مقدار undefined بدهد. | متغیر let، قابلیت Hoisting ندارد. | متغیر let، قابلیت Hoisting ندارد. |
| متغیرهای var یا دارای حوزه سراسری (Global Scope) یا حوزه تابعی (Function Scope) هستند. | متغیرهای let دارای حوزه بلاکی (Block Scope)  هستند. | ثابت‌های const دارای حوزه بلاکی (Block Scope) هستند. |

***<!—Q8 -->***

|  |  |  |
| --- | --- | --- |
|  | توضیحات | مثال |
| library | مفهوم کتابخانه در برنامه نویسی نیز شبیه به کتابخانه معمولی است. یعنی کتابخانه مجموعه‌ای از کلاس‌ها و توابع و کدهای آماده است که باعث می‌شود به جای اینکه این کدها را از اول خودمان بنویسیم، از کدهای آماده‌ی از قبل نوشته شده استفاده کنیم تا کارمان آسان شود و درگیر کدهای پیچیده و وقت‌گیر نشویم. | JQUERY  و  Sort table |
| framework | Framework (فریمورک) در فارسی به معنی چهارچوب است. در حقیقت فریم‌ورک‌ها یک سری چهارچوب و اسکلت هستند که از پیش آماده هستند و ما برای تکمیل این اسکلت‌ها، کدهای خودمان را به آن اضافه می‌کنیم. در حقیقت شباهت کتابخانه با فریمورک در همین آسان‌تر کردن کدها خلاصه می‌شود. | Django  و  Laravel |
| differences | کتابخانه مجموعه‌ای از کدهای از پیش نوشته شده است که در صورت لزوم از آن برای راحتی کار خود و صرفه‌جویی وقت، استفاده می‌کنیم. در مقابل، فریم‌ورک در واقع چهارچوب و اسکلت از پیش ساخته شده کدها است که در اختیار برنامه نویس قرار داده شده است. برنامه نویس با اضافه کردن کدهای خود به این چهارچوب، برنامه خود را می‌سازد. |  |

***<!—Q9 -->***

1 **<head>**

The <head> section is located at the beginning of the HTML document, before the actual content of the page.

It contains meta-information about the document, such as the title of the page, links to external stylesheets, metadata, character set declarations, and references to scripts.

2 <body>

The <body> section contains the actual content of the HTML document, such as text, images, links, forms, and other elements that make up the visible part of the webpage.

This is where you include the elements that users interact with and see on the web page

***<!—Q10 -->***

const person = {

    userfirstName : prompt( "What's your First Name ? "),

    userlastName : prompt( " What's your Last Name ? " )

};

alert(`First Name: ${person.userfirstName}\nLast Name: ${person.userlastName}`);

***<!—Q11 -->***

alert("Hello \"Maktab\"");