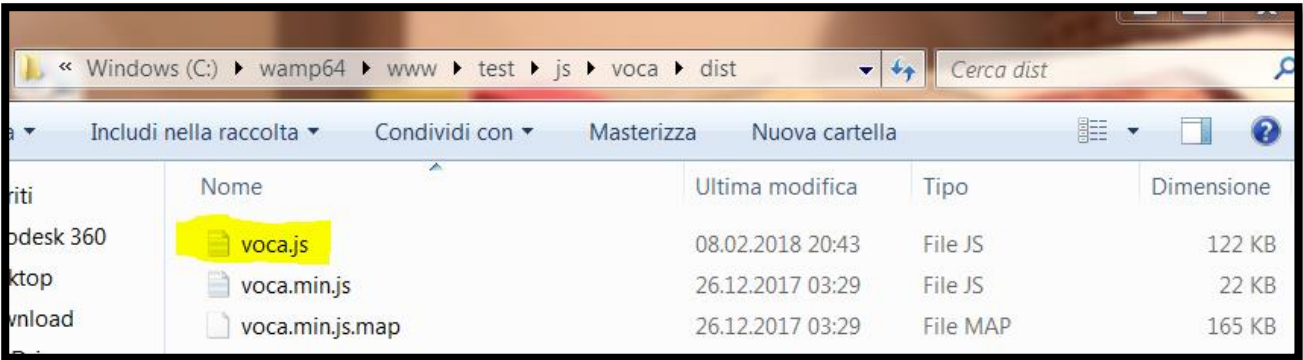


File & location:



| Nome | Ultima modifica | Tipo | Dimensione |
|-----------------|------------------|----------|------------|
| voqa.js | 08.02.2018 20:43 | File JS | 122 KB |
| voqa.min.js | 26.12.2017 03:29 | File JS | 22 KB |
| voqa.min.js.map | 26.12.2017 03:29 | File MAP | 165 KB |

Script including in test file (.../www/test/test.php)

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=8" />
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css">
<script type="text/javascript" charset="utf8" src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.8.2.min.js"></script>
<script src="./js/voqa/dist/voqa.js"></script>

<title>test</title>
<script type="text/javascript">
```

Vprintf call in test file (.../www/test/test.php)

```
function populate(item, index){
    var shuffle=true;
    var answers=item[index].answers;
    var answekeys=Object.keys(answers);
    if(shuffle){answekeys=answekeys.sort(function () {return (Math.round(Math.random()-0.5));});}

    $('#questions').html('');

    //i=loop number; key=shuffled or not array keys
    $.each(answekeys, function(i,key) {

        var text=answers[key].text;

        if(answers[key].behaviour=='dynamic'){

            var terms=Object.values(answers[key].vars).sort(function () {return (Math.round(Math.random()-0.5));});

            text=v.vprintf(text, terms);

        }

        $('#questions').prepend("<input type='radio' name='radio' type='"+key+"'>"+text+"</input>");
        $('#questions').prepend("<br>");

    });

    $('#questions').prepend('<b>'+item[index].question+'</b><br>');
}

//populate
```

Data array in test file, then passed to javascript in json format (.../www/test/test.php)

```
$item[1]['answers'][2]=array();  
$item[1]['answers'][2]['behaviour']="dynamic";  
$item[1]['answers'][2]['vars']=array('random1', 'random2', 'random3', 'random4');  
$item[1]['answers'][2]['text']="Risposta errata con variabile: %s";  
$item[1]['answers'][2]['message']=null;
```

Result in Chrome

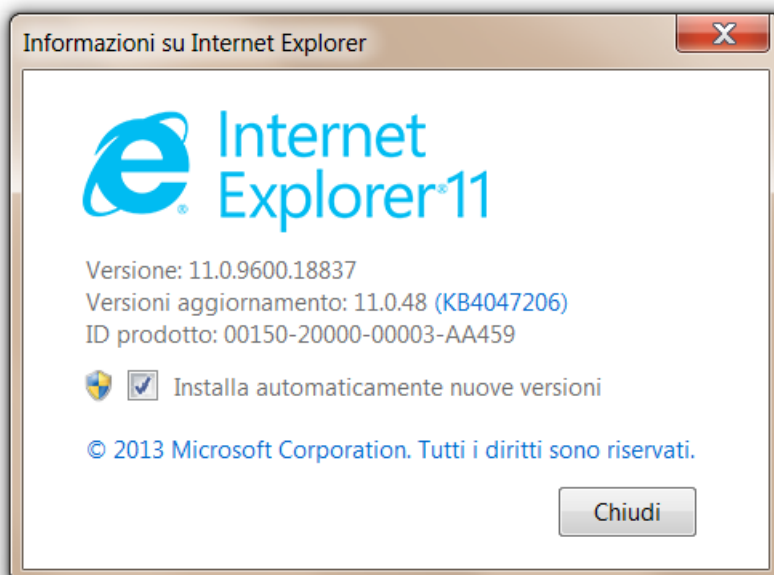
Domanda 2

- Risposta errata con variabile: random3
- risposta esatta
- risposta errata

Submit

button

Explorer version



Explorer output

Here comes the form

Submit

button

Inspecting voca.js with debugger (keys method not supported)

```
2026 '&': /(&amp;)|(&#x0*26;)|(&#0*38;)/gi,  
2027 "'": /(&quot;)|(&#x0*22;)|(&#0*34;)/gi,  
2028 """: /(&#x0*27;)|(&#0*39;)/gi,  
2029 ```: /(&#x0*60;)|(&#0*96;)/gi  
2030 };  
2031 var characters = Object.keys(unescapeCharactersMap);  
L'oggetto non supporta la proprietà o il metodo 'keys'  
2032  
2033 /**  
2034 * Replaces the HTML entities with corresponding characters
```

Voca.js original file (unmodified)

```
/*!  
 * Voca string library 1.4.0  
 * https://vocajs.com  
 *  
 * Copyright Dmitri Pavlutin and other contributors  
 * Released under the MIT license  
 */  
|  
  
(function (global, factory) {  
  typeof exports === 'object' && typeof module !== 'undefined' ? module.exports = factory() :  
  typeof define === 'function' && define.amd ? define(factory) :  
  (global.v = factory());  
})(this, (function () { 'use strict';  
  
/**  
 * Checks if `value` is `null` or `undefined`  
 *  
 * @ignore  
 * @function isNil  
 * @param {*} value The object to check  
 * @return {boolean} Returns `true` if `value` is `undefined` or `null`, `false` otherwise  
 */
```

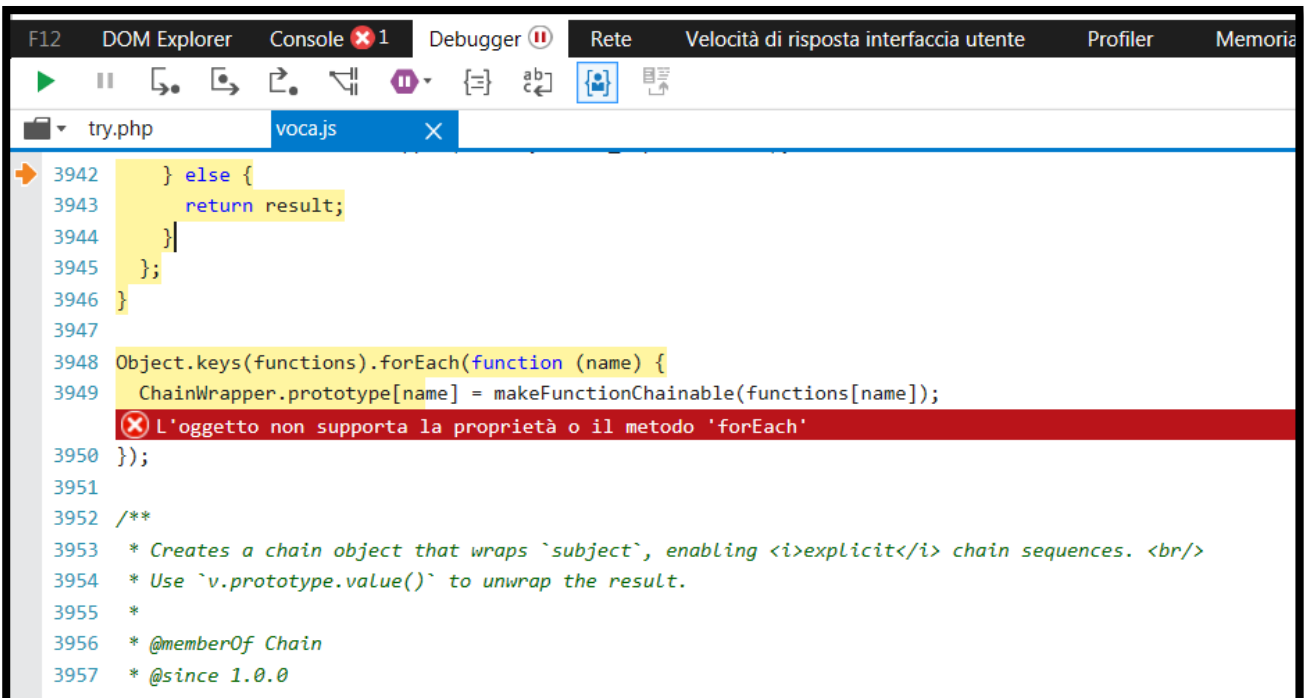
First add

```
/*!
 * Voca string library 1.4.0
 * https://vocajs.com
 *
 * Copyright Dmitri Pavlutin and other contributors
 * Released under the MIT license
 */
|
|-----add 1
if (!Object.keys) {
  Object.keys = function(obj) {
    var keys = [];
    for (var i in obj) {
      if (obj.hasOwnProperty(i)) {
        keys.push(i);
      }
    }
    return keys;
  };
}

(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? module.exports = factory() :
  typeof define === 'function' && define.amd ? define(factory) :
  (global.v = factory());
})(this, (function () { 'use strict';

/**
 * Checks if `value` is `null` or `undefined`
 *
 * @ignore
 */
```

Reload and check next error (forEach method not supported)



```
F12  DOM Explorer  Console 1  Debugger  Rete  Velocità di risposta interfaccia utente  Profiler  Memoria
try.php  vocajs
3942   } else {
3943     return result;
3944   }
3945 };
3946 }
3947
3948 Object.keys(functions).forEach(function (name) {
3949   ChainWrapper.prototype[name] = makeFunctionChainable(functions[name]);
3950 });
3951
3952 /**
3953  * Creates a chain object that wraps `subject`, enabling <i>explicit</i> chain sequences. <br/>
3954  * Use `v.prototype.value()` to unwrap the result.
3955  *
3956  * @memberOf Chain
3957  * @since 1.0.0
```

Add 2

```
*/
//-----add 1
if (!Object.keys) {
  Object.keys = function(obj) {
    var keys = [];
    for (var i in obj) {
      if (obj.hasOwnProperty(i)) {
        keys.push(i);
      }
    }
    return keys;
  };
}

//-----add 2
if (!Array.prototype.forEach) {
  Array.prototype.forEach = function(fn, scope) {
    for(var i = 0, len = this.length; i < len; ++i) {
      fn.call(scope, this[i], i, this);
    }
  }
}

(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? module.exports = factory() :
  typeof define === 'function' && define.amd ? define(factory) :
  (global.v = factory());
})(this, (function () { 'use strict';

/**
 * Checks if `value` is `null` or `undefined`
 *

```

Reload and check next error (values method not supported)

```
try.php x vocajs
93 // i=loop number, key=shuffled or not array keys
94 $.each(answerskeys, function(i,key) {
95
96     var text=answers[key].text;
97
98     if(answers[key].behaviour=='dynamic'){
99
100         var terms=Object.values(answers[key].vars).sort(function (){return (Math.round(Math.random())-0.5)});
101         text=v.vprintf(text, terms);
102     }//if dynamic
103
104     $('#questions').prepend("<input type='radio' name='radio' type='"+key+"'>"+text+"</input>");
105     $('#questions').prepend("<br>");
106
107 });//each
108
109
110 $('#questions').prepend('<b>'+item[index].question+'</b><br>');
```

L'oggetto non supporta la proprietà o il metodo 'values'

Add 3

```
},
}

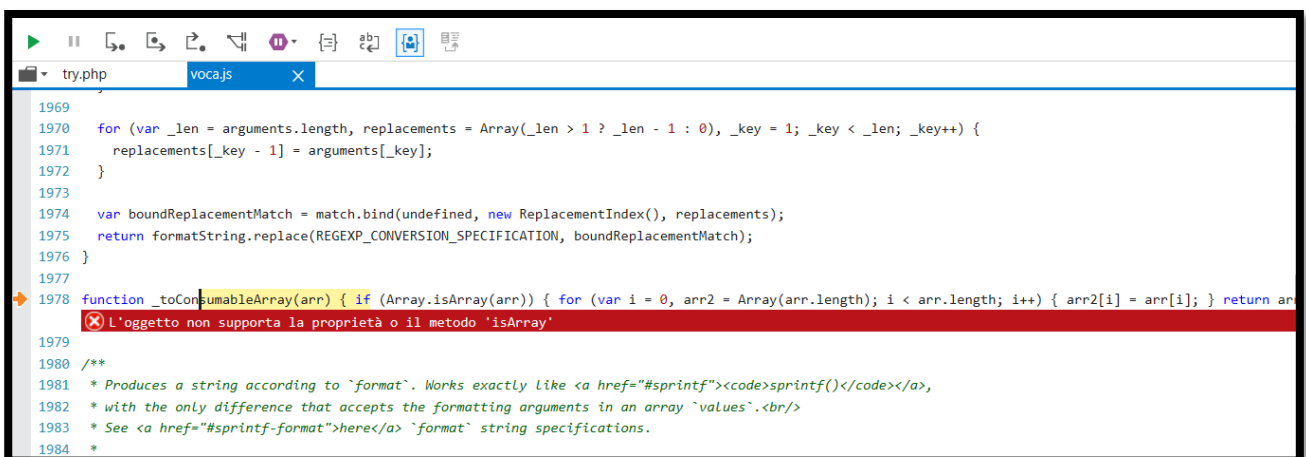
//-----add 2
if (!Array.prototype.forEach) {
  Array.prototype.forEach = function(fn, scope) {
    for (var i = 0, len = this.length; i < len; ++i) {
      fn.call(scope, this[i], i, this);
    }
  }
}

//-----add 3
if (!Object.values) {
  Object.values = function(obj) {
    var res = [];
    for (var i in obj) {
      if (obj.hasOwnProperty(i)) {
        res.push(obj[i]);
      }
    }
    return res;
  };
}

(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? module.exports = factory() :
  typeof define === 'function' && define.amd ? define(factory) :
  (global.v = factory());
})(this, (function () { 'use strict';

/**
```

Reload and check next error (isArray method not supported)



```
1969
1970   for (var _len = arguments.length, replacements = Array(_len > 1 ? _len - 1 : 0), _key = 1; _key < _len; _key++) {
1971     replacements[_key - 1] = arguments[_key];
1972   }
1973
1974   var boundReplacementMatch = match.bind(undefined, new ReplacementIndex(), replacements);
1975   return formatString.replace(REGEXP_CONVERSION_SPECIFICATION, boundReplacementMatch);
1976 }
1977
1978 function _toConsumableArray(arr) { if (Array.isArray(arr)) { for (var i = 0, arr2 = Array(arr.length); i < arr.length; i++) { arr2[i] = arr[i]; } return arr2; } else { return Array.from(arr); } }
1979
1980 /**
1981  * Produces a string according to `format`. Works exactly like printf(),
1982  * with the only difference that accepts the formatting arguments in an array `values`.
1983  * See printf-format here for `format` string specifications.
1984  *
```

Add 4

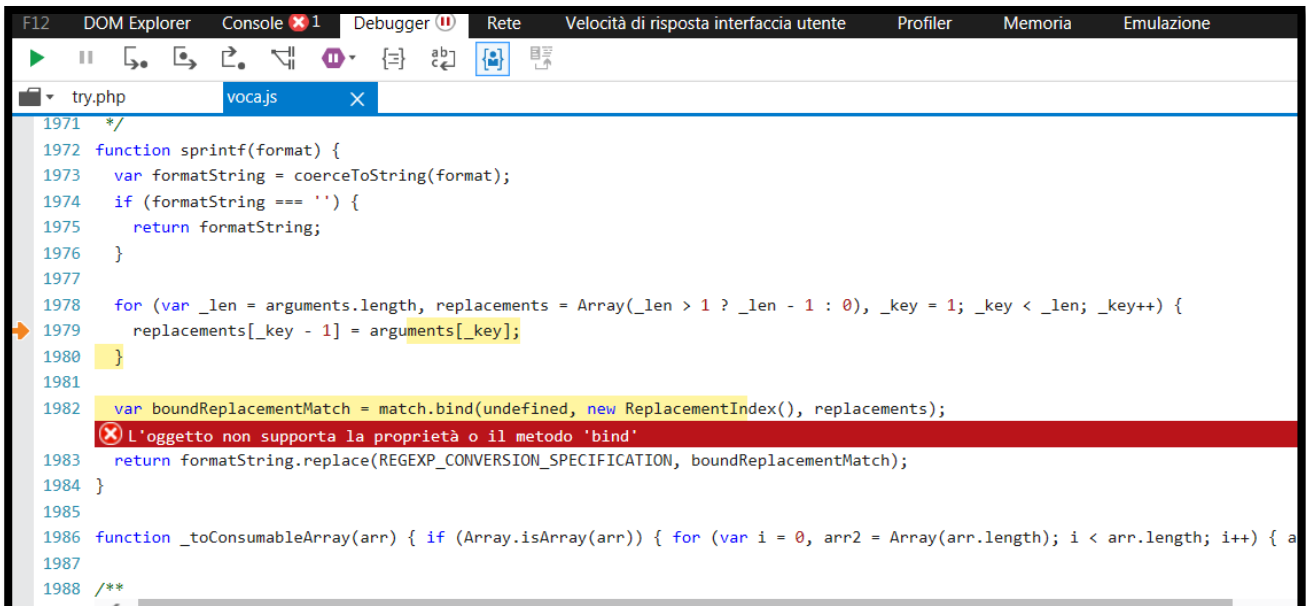
```
    }
  }
}

//-----add 3
if (!Object.values) {
  Object.values = function(obj) {
    var res = [];
    for (var i in obj) {
      if (obj.hasOwnProperty(i)) {
        res.push(obj[i]);
      }
    }
    return res;
  };
}

//-----add 4
if (!Array.isArray) {
  Array.isArray = function(item) {
    if(item instanceof Array){return true;}
    return false;
  }
}

(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? module.exports = factory() :
  typeof define === 'function' && define.amd ? define(factory) :
  (global.v = factory());
})(this, function () { 'use strict';
```

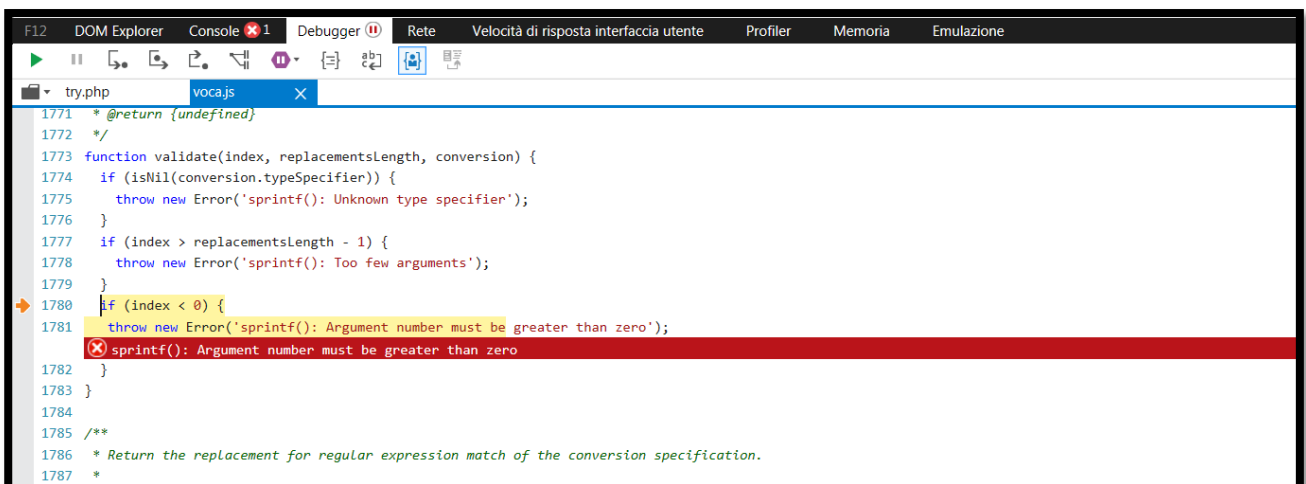
Reload and check next error (bind method not supported)



Add 5

```
}  
  
//-----add 5  
if (!Function.prototype.bind) {  
  Function.prototype.bind = function(oThis) {  
    if (typeof this !== 'function') {  
      // closest thing possible to the ECMAScript 5  
      // internal IsCallable function  
      throw new TypeError('Function.prototype.bind - what is trying to be bound is not callable');  
    }  
  
    var aArgs = Array.prototype.slice.call(arguments, 1),  
        fToBind = this,  
        fNOP = function() {},  
        fBound = function() {  
          return fToBind.apply(this instanceof fNOP && oThis  
            ? this  
            : oThis,  
            aArgs.concat(Array.prototype.slice.call(arguments)));  
        };  
  
    fNOP.prototype = this.prototype;  
    fBound.prototype = new fNOP();  
  
    return fBound;  
  };  
}  
  
(function (global, factory) {  
  typeof exports === 'object' && typeof module !== 'undefined' ? module.exports = factory() :  
  typeof define === 'function' && define.amd ? define(factory) :  
  (global.v = factory());  
})(this, (function () { 'use strict';  
  
/**  
 * Checks if `value` is `null` or `undefined`  
 *  
 */
```

Reload and check next error: here an error is thrown by the **validate** function



```
F12  DOM Explorer  Console 1  Debugger  Rete  Velocità di risposta interfaccia utente  Profiler  Memoria  Emulazione  
try.php  voca.js  
1771  * @return {undefined}  
1772  */  
1773  function validate(index, replacementsLength, conversion) {  
1774    if (isNil(conversion.typeSpecifier)) {  
1775      throw new Error('sprintf(): Unknown type specifier');  
1776    }  
1777    if (index > replacementsLength - 1) {  
1778      throw new Error('sprintf(): Too few arguments');  
1779    }  
1780    if (index < 0) {  
1781      throw new Error('sprintf(): Argument number must be greater than zero');  
1782    }  
1783  }  
1784  }  
1785  /**  
1786  * Return the replacement for regular expression match of the conversion specification.  
1787  */
```


The **validate** function is called inside the **match** function using **actualReplacementIndex** as argument.

```

* @param {string} precisionSpecifier The precision specifier says how many decimal digits should be displayed for floats.
* @param {string} typeSpecifier The type specifier says what type the argument data should be treated as.
* @return {string} Returns the computed replacement.
*/
function match(replacementIndex, replacements, conversionSpecification, percent, position, signSpecifier, paddingSpecifier, alignmentSpecifier) {
    var conversion = new ConversionSpecification({
        percent: percent,
        signSpecifier: signSpecifier,
        paddingSpecifier: paddingSpecifier,
        alignmentSpecifier: alignmentSpecifier,
        width: coerceToNumber(widthSpecifier, null),
        precision: coerceToNumber(precisionSpecifier, null),
        typeSpecifier: typeSpecifier
    });
    if (conversion.isPercentLiteral()) {
        return conversionSpecification.slice(1);
    }
    var actualReplacementIndex = replacementIndex.getIndexByPosition(position);
    replacementIndex.incrementOnEmptyPosition(position);

    validate(actualReplacementIndex, replacements.length, conversion);
    return compute(replacements[actualReplacementIndex], conversion);
}

/**
 * Produces a string according to `format`.
 *
 * <div id="sprintf-format" class="smaller">
 * `format` string is composed of zero or more directives: ordinary characters (not <code>%</code>), which are copied
 */

```

The **var actualReplacementIndex** contains value -1

```

function match(replacementIndex, replacements, conversionSpecification, percent, position, signSpecifier, paddingSpecifier, alignmentSpecifier) {
    var conversion = new ConversionSpecification({
        percent: percent,
        signSpecifier: signSpecifier,
        paddingSpecifier: paddingSpecifier,
        alignmentSpecifier: alignmentSpecifier,
        width: coerceToNumber(widthSpecifier, null),
        precision: coerceToNumber(precisionSpecifier, null),
        typeSpecifier: typeSpecifier
    });
    if (conversion.isPercentLiteral()) {
        return conversionSpecification.slice(1);
    }
    var actualReplacementIndex = replacementIndex.getIndexByPosition(position);
    replacementIndex.incrementOnEmptyPosition(position);

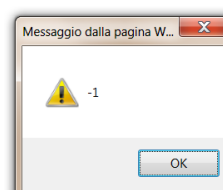
    //testing
    alert(actualReplacementIndex);
    |
    validate(actualReplacementIndex, replacements.length, conversion);
    return compute(replacements[actualReplacementIndex], conversion);
}

```

- risposta errata
- risposta esatta

Submit

button



The `var actualReplacementIndex` is defined here

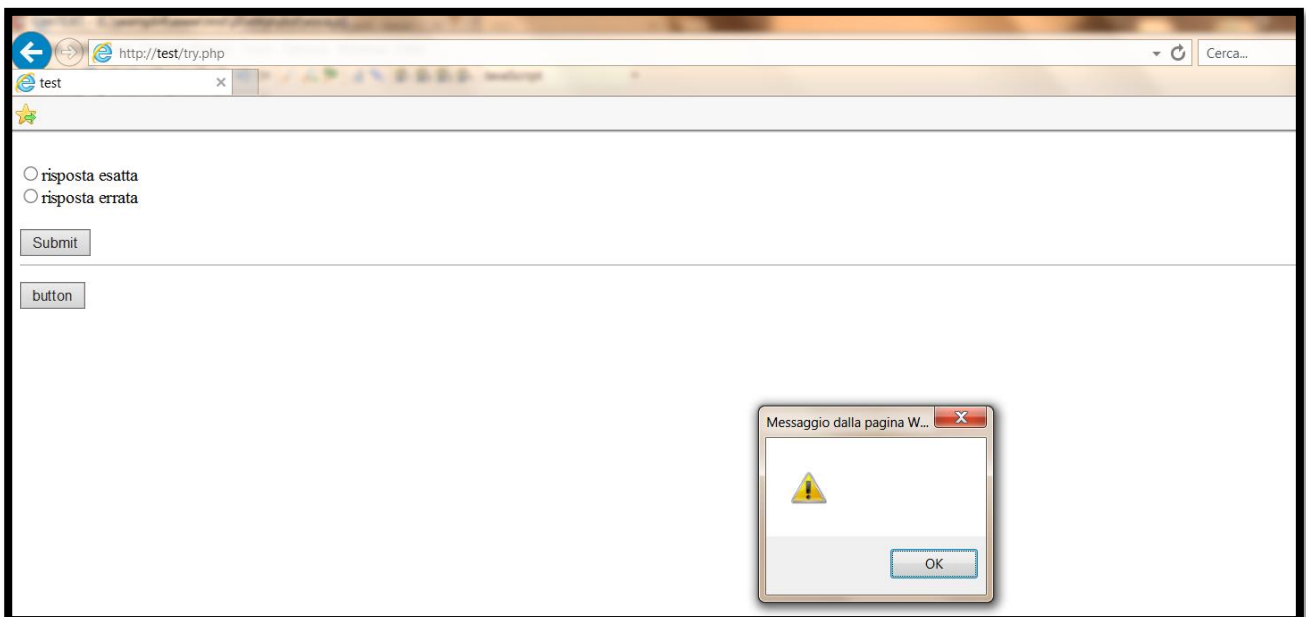
```
* @param {number} [position] The replacement position.
* @return {number} The replacement index.
*/
ReplacementIndex.prototype.getIndexByPosition = function (position) {
    return isNil(position) ? this.index : position - 1;
};

// Type specifiers
var TYPE_INTEGER = 'i';
```

The `position` argument contains nothing (not `null` nor `undefined`) and it is passed to the `isNil` function

```
ReplacementIndex.prototype.getIndexByPosition = function (position) {

    //test
    alert(position);
    |
    return isNil(position) ? this.index : position - 1;
};
```



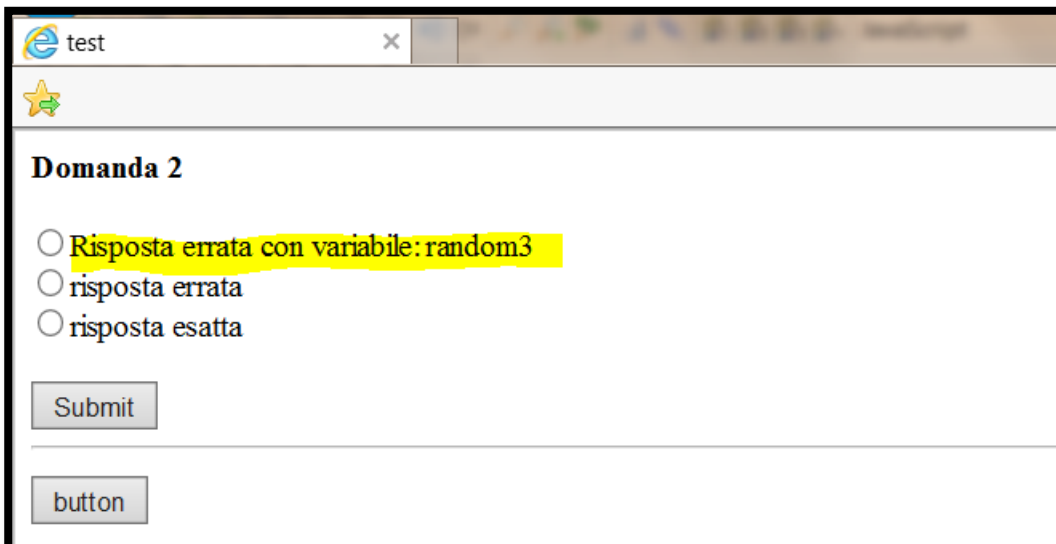
The `isNil` function returns `true` only if the argument is `null` or `undefined`

```
/**
 * @param {*} value the object to check
 * @return {boolean} Returns `true` if `value` is `undefined` or `null`, `false` otherwise
 */
function isNil(value) {
  return value === undefined || value === null;
}
```

Adding the case `nothing` (empty string)...

```
/**
 * @param {*} value the object to check
 * @return {boolean} Returns `true` if `value` is `undefined` or `null`, `false` otherwise
 */
function isNil(value) {
  return value === undefined || value === null || value === '';
}
```

...then the substitution works



The screenshot shows a browser window with a single tab titled "test". Below the address bar, there is a star icon and a question titled "Domanda 2". The question has three radio button options: "Risposta errata con variabile: random3", "risposta errata", and "risposta esatta". The first option is highlighted in yellow. Below the options are two buttons: "Submit" and "button".