Data Transfer from HTML to JavaScript

(由 HTML 到 JavaScript (JS) 的数据传递)

HTML is a markup language, not a programming language. HTML itself cannot do any math calculation, even as simple as 1 + 2. Usually, a user inputs data using HTML and then the data are transferred to PHP, a server-side programming language, for processing, for example:

```
<html>
<form action = “processing.php">
        <input type="radio" id="single" name="status" value="single"> Single Return<br>
        <input type="radio" id="joint" name="status" value="joint"> Joint Return<br>
        <input type="button" onclick="chk()" value="continue">
        <input type="submit" value="Submit">
</form>
</html>
```

After user inputs all necessary data and then clicks the “Submit” button, the data will be, then, transferred to server, and there a program processing.php will perform all predefined processing. Then, question arises: is it possible to transfer data to the client-side JS for processing, rather than to transfer to remote server? Off cause, some type of job is not appropriated for security reasons, such as user login. However for most other type tasks, using JS is more convenient.

I don’t think I learned this type of topics previously and determined to try to solve it now. After some struggling, limited success finally accomplished, and data transferring from HTML to JS cab be easily done for four types of data, as summarized below,

1. String

```
<html>
        <p>Enter No.1 <input type = "text" name = "" id = "st1"></p>
        <p>Enter No.2 <input type = "text" name = "" id = "st2"></p>
        <p>Enter No.3 <input type = "text" name = "" id = "st3"></p>
        <button onclick = "prcs()">Continue</button>
        <p id="out"><br> </p>
</html>
<script>
        function prcs()   //After the user clicks the “Continue” button, the JS function “prcs()” will be
                          // called to perform data processing defined in the function.
        {
                var str = new Array();              //declare an array with the name of “str”
```

```
            str[1] = document.getElementById("st1").value;
            str[2] = document.getElementById("st2").value;
            str[3] = document.getElementById("st3").value;

        var i;
        for (i = 1; i <= 3; i++)
        {
            o("str  " + i + " = " + str[i] + "<br>");
        }
  </script>
```

2. Numerical number

```
<html>
        <p>Enter No.1 <input type = "number" name = "" id = "no1"></p>
        <p>Enter No.2 <input type = " number" name = "" id = "no2"></p>
        <p>Enter No.3 <input type = " number" name = "" id = "no3"></p>
          //though “type = "number" is specified, after transferring to JS, it is still a string!
        <button onclick = "prcs()">Continue</button>
        <p id="out"><br> </p>
</html>
<script>
          function prcs()
          {
            var No = new Array();     //declare an array with the name of “No”

            No[1] = parseFloat (document.getElementById("no1").value);
            No[2] = parseFloat (document.getElementById("no2").value);
            No[3] = parseFloat (document.getElementById("no3").value);
              //data transferred from HTML are strings, parseFloat() function must be used to
              // convert them to (floating point) numbers.

            var i;
            for (i = 1; i <= 3; i++)
            {
                o("No  " + i + " = " + No[i] + "<br>");
        }
  </script>
```

3. Radio button

```
<html>
<input type="radio" id="single" name="status" value="single"> Single Return<br>
<input type="radio" id="joint" name="status" value="joint"> Joit Return<br>
<input type="button" onclick="chk()" value="continue">
<!-- radio buttons share the same name make one group and only one button can be checked.
-->
</html>

<script>
function chk()
{
var radio1 = document.getElementById("single").checked;
var radio2 = document.getElementById("joint").checked;
// it was wrong to use "var radio1 = document.getElementById("single").value;"

if (radio1 == false && radio2 == false)
{
        alert("You must select your filing status! Please start over.");

        document.getElementById("continue").onclick = function ()
        {
                location.href = "html_radio_js.html";
        }
  }

  if (radio1 == true) {
     prt("Single filing.");
  }

  if (radio2 == true) {
     prt("joint filing.");
  }
}

//checked = on (.value)

</script>
```

4. Check box

```
<html>
<form>
    <input type="checkbox" id="chk1" >checkbox 1<br>
```

```
    <input type="checkbox" id="chk2" >checkbox 2<br><br>
    <input type="button" onclick="chk()" value="continue">
  </form>
  <p id = "out"></p>
</html>
 <script>
    function o(opt)
    {
       myOutput = document.getElementById("out");  //"Id" is the same as pre-defined in the HTML code above in <p>.
       myOutput.innerHTML += opt;     //1."myOutput" can be any identifier. 2. it is important to use "+=", not "=".
    }

    function chk() {
       var chkbox1 = document.getElementById("chk1").checked;
       var chkbox2 = document.getElementById("chk2").checked;

       //var chkbox1 = document.getElementById("chk1".value);
       //var chkbox2 = document.getElementById("chk2").value;
       //checked = null; uncheck = on
       //still did not work out the code that makes it work.

       if (chkbox1 == true){
          o("chkbox1 is checked.<br>")
       }
       else {
           o("chkbox1 is NOT checked.<br>")
       }

       if (chkbox2 == true) {
          o("chkbox2 is checked.<br>")
       }
       else {
           o("chkbox2 is NOT checked.<br>")
       }
          o("<br>Great! A new output function using only \"o\" has been created!<br><br>");
    }
  </script>
```

5. Data file – to be continued