

HYPHER-DUPLET WEB PAGE

Duplets ask a viewer to make a comparison between a pair of pictures. It is possible to construct an interactive electronic display to present a series of duplets where each side can be advanced separately. This provides the opportunity to make a series of comparisons that interconnects a whole range of pictures, not just specific pairs.

This project provides the means to present such a series of duplets where the individual pictures are independent, or more specifically hyper-dependent. This requires that each left-hand picture resonate with each right-hand picture, and vice versa.

The overall procedure for creating this interactive presentation is as follows:

- Create the Pix
- Redesign the Web Page
- Replace the First 2 Pix to the Web Page (don't delete, just replace)
- Alter the JavaScript for the other Pix
- Hand in the Folder with Pix and Page

CREATE the PICTURES

The task here is to define topic for yourself and then collect pictures that discuss that topic. A duplet requires that two pictures sit next to each other, and that requires that two pictures relate on more than one level, like maybe in a formal (visual) way, and in an emotional way, and they should share some idea. Their sense of space and time and light should also correspond.

With hyper-duplets, the relationships must extend beyond single pairs of pictures, so that all pictures in a set could be on either side of a duplet. This interactive display will allow the deck to be shuffled by your audience, so you have to be consistent.

PICTURE SIZES

All pictures in the slide show must be the same size and the same orientation. HTML only allows changing the source picture file, not the size specs.

There are 1440 x 900 pixels available across the entire screen of a typical 15" laptop computer. This provides about 1200 x 700 pixels in width after leaving room for the 'chrome' of the browser window (toolbars, etc.) and some border space. When this space is split into two parts to hold a horizontal duplet, the individual pictures will be 600 pixels wide by 400 high. Even if the pictures will be a vertical duplet, use this size so the entire site is consistent.

resize _____ all duplet web pix to 600 x 400 (or 500 square)
use Batch Resize recipe for Bridge+ Photoshop Image Processor (below)
save for web use jpeg, quality =10, convert to sRGB
use Batch Rename recipe for Bridge (below)
use the name format [picL1.jpg](#), [picL2.jpg](#),... [picL20.jpg](#)
and the names format [picR1.jpg](#), [picR2.jpg](#),... [picR20.jpg](#)
the spelling must be **exactly** as above
to match the coding in the javascript web page

DESIGN the WEB PAGE

A custom Hyper-Duplet web page template has been constructed for you in DreamWeaver. Special JavaScript code is included to change each picture when a viewer clicks on it.

FILE SETUP

create _____ a folder for the hyper-duplet piece
the entire folder will be uploaded into the OWLbox folder
use the names of the 2 partners, e.g. [kristen_rosieHD](#)
get a copy _____ of the hyper-duplet html page
from either the instructor's web site, or the class web folder
save as _____ immediately, renamed as e.g. [kristen_rosieHD.html](#)
into the folder

PAGE ALTERATION

The Artist's Name and the Page Title must be altered to *your names*
open _____ the hyper-duplet template HTML document
alter _____ the Artist's Names to the real artist's name
save as _____ again as above...

PICTURE COLLECTION

Move all of your picture files into the hyper-duplet folder. This can be done when you are rescaling, optimizing, and saving for the web, or you can create them first and move them into the appropriate folders later.

See the detailed Recipe Pages for more details, or follows this quick guide:

collect _____ your pictures into the appropriate folders
open _____ Bridge
 use _____ Batch Rename
rename _____ the left-hand pix for each set (exactly):
 [picL1.jpg](#), [picL2.jpg](#), [picL3.jpg](#), . . . [picL19.jpg](#), [picL20.jpg](#)
rename _____ the right-hand pix for each set set (exactly):
 [picR1.jpg](#), [picR2.jpg](#), [picR3.jpg](#), . . . [picR19.jpg](#), [picR20.jpg](#)

select _____ all pictures
open _____ into Photoshop Image Processor from within Bridge
file type _____ use JPEG, quality 10, convert to sRGB
resize _____ to 600 x 600 pixels (or 500 pixels square for medium-format shots)
 the aspect ratio will be retained, so 2:3 pix will be 600 x 400, vertical or horizontal

The HTML page will need no other alterations than this, as long as you have the filenames spelled correctly. That's it!

If you have some number of pictures other than the 20 for which the demo page is set up for, then you will have to add one piece of code for each added picture, or subtract one piece of code for each photo subtracted. See the explanation below for further details.

HTML EXPLANATION

PICTURE DEFINITION

The HTML code for a picture on a web page looks like this:

```

```

The tag is an IMAGE tag and it includes the name of the SOURCE file, the WIDTH and HEIGHT of the file, a NAME for the picture object, and here, the BORDER is set to 1.

HTML and JAVASCRIPT EXPLANATION

When DreamWeaver places a picture into the page, that filename is automatically written into the page code. In this page the picture is placed inside a <div> tag to center it, then inside an <a> tag to link it as shown below:

```
<div align="right"><a href="#" onClick="nextLeftPic()">
</a></div>
```

The href= property defines a link, but here it is an empty place holder. This is the way DreamWeaver sets up its JavaScript references.

The onClick= is the actual event that triggers the action, and the "nextLeftPic()" is the name of the JavaScript function that is called when the event occurs.

The size of the pictures will be set to the sizes specified here. The NAME of the picture that appears on the far left side of the PROPERTIES INSPECTOR must always be either leftPic or rightPic for this particular JavaScript to work.

An onBlur() function has been added to cancel out the addition of an annoying border that Internet Explorer draws around the active link.

JAVASCRIPT EXPLANATION

Javascript has the ability to swap picture files. This means that when one picture sits on a web page, with certain attributes such as size and border, the name of the source file can be swapped. This means that a different pictures can replace the first. This is how Rollovers work, however Rollovers swap the pictures when the mouse moves over the graphic, and usually replace the original picture when the mouse moves away from the picture.

In the case of the hyper-duplet script, each successive click on the picture swaps in another picture, and so on, in a looping manner.

```
onClick="nextLeftPic()
```

is where the action of clicking the mouse calls the script called 'nextLeftPic'.

The javascript uses the filenames stored in the array called 'allLeftPix'

```
allLeftPix =
newArray("picl1.jpg", "picl2.jpg", "picl3.jpg", "picl4.jpg", "picl5.jpg",
"picl6.jpg", "picl7.jpg", "picl8.jpg", "picl9.jpg", "picl10.jpg",
"picl11.jpg", "picl12.jpg", "picl13.jpg", "picl14.jpg", "picl15.jpg",
"picl16.jpg", "picl17.jpg", "picl18.jpg", "picl19.jpg", "picl20.jpg")
```

And the same for the Right Hand pictures...

The JAVASCRIPT FUNCTIONS

Here is an explanation of how the Javascript functions work, line by line:

The actual JavaScript starts in the head section of the HTML page with the tag:

```
<script language="JavaScript">
<!--
```

The `arrays` containing the names of the pictures in the slide show are defined with these lines as described earlier. You must spell your filenames exactly as they are here:

```
allLeftPix = new
    Array("picL1.jpg","picL2.jpg","picL3.jpg","picL4.jpg","picL5.jpg",
    "picL6.jpg","picL7.jpg","picL8.jpg","picL9.jpg","picL10.jpg",
    "picL11.jpg","picL12.jpg","picL13.jpg","picL14.jpg","picL15.jpg",
    "picL16.jpg","picL17.jpg","picL18.jpg","picL19.jpg","picL20.jpg")
allRightPix = new
    Array("picR1.jpg","picR2.jpg","picR3.jpg","picR4.jpg","picR5.jpg")
    "picR6.jpg","picR7.jpg","picR8.jpg","picR9.jpg","picR10.jpg",
    "picR11.jpg","picR12.jpg","picR13.jpg","picR14.jpg","picR15.jpg",
    "picR16.jpg","picR17.jpg","picR18.jpg","picR19.jpg","picR20.jpg")
```

The counters are set to zero with these 2 lines:

```
nowLeft = 0
nowRight = 0
```

This line defines the name of the function:

```
//
function nextLeftPic() {
```

This line increments the counter, so the proper frame is being shown:

```
nowLeft++
```

This line checks to see if the counter is bigger than the number of pictures in the slide show. If it is, the counter is reset to zero, and the first picture is shown again:

```
if (nowLeft >= allLeftPix.length) {nowLeft = 0}
```

This line changes the source of the JPEG picture object called `leftPic` to the name of the file you have specified in the array definition lines:

```
document.leftPic.src = allLeftPix[nowLeft]
```

The function called `nextRightPic()` works exactly the same way, except that it works on the right picture instead of the left picture:

The script ends with the lines:

```
//-->
</script>
```

The COMPLETE HTML CODE / HEAD SECTION

Below is the code for the head section of the page. This includes the array definitions, the javascript functions, and a style sheet to define the look of the text on the page.

```
<html>
<head>
<title>your name hyper-duplet</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript">
<!--
allLeftPix = new
    Array("picL1.jpg","picL2.jpg","picL3.jpg","picL4.jpg","picL5.jpg",
    "picL6.jpg","picL7.jpg","picL8.jpg","picL9.jpg","picL10.jpg",
    "picL11.jpg","picL12.jpg","picL13.jpg","picL14.jpg","picL15.jpg",
    "picL16.jpg","picL17.jpg","picL18.jpg","picL19.jpg","picL20.jpg")
allRightPix = new
    Array("picR1.jpg","picR2.jpg","picR3.jpg","picR4.jpg","picR5.jpg")
    "picR6.jpg","picR7.jpg","picR8.jpg","picR9.jpg","picR10.jpg",
    "picR11.jpg","picR12.jpg","picR13.jpg","picR14.jpg","picR15.jpg",
    "picR16.jpg","picR17.jpg","picR18.jpg","picR19.jpg","picR20.jpg")
nowLeft = 0
nowRight = 0
//
function nextLeftPic() {
    nowLeft++
    if (nowLeft >= allLeftPix.length) {nowLeft = 0}
    document.leftPic.src = allLeftPix[nowLeft]
}
//
function nextRightPic() {
    nowRight++
    if (nowRight >= allRightPix.length) {nowRight = 0}
    document.rightPic.src = allRightPix[nowRight]
}
//-->
</script>
<style type="text/css">
<!--
.artistNameHD { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
    16px; font-weight: bold; color: #FFFFFF}
-->
</style>
</head>
```

The HTML CODE / BODY SECTION

Below is the body section of the code. This is where the tables are defined, containing the pictures that are connected to the javascript events.

```
<body bgcolor="#000000" text="#FFFFFF" link="#CCCCFF" vlink="#FFCCCC"
  alink="#CCCCCC">
<table border="0" align="center" width="1200" cellpadding="0" cellspacing="0">
  <tr>
    <td>
      <div align="center" class= artistNameHD">artist name
      <p>&nbsp;</p>
      </div>
    </td>
  </tr>
  <tr>
    <td>
      <div align="right">
        <table width="788" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td>
              <div align="left"><a href="#"
onClick="blur(),nextLeftPic()">
                </a></div>
            </td>
            <td>&nbsp;</td>
            <td width="20">
              <div align="right"><a href="#"
onClick="blur(),nextRightPic()">
                </a></div>
            </td>
          </tr>
        </table>
      </div>
    </td>
  </tr>
  <tr>
    <td>
      <p align="center">&nbsp;</p>
      <div align="center"></div>
    </td>
  </tr>
</table>
</body>
</html>
```

It is possible to copy this code directly into DreamWeaver, or better, open the copy of this page that is on the website www.berk-edu.com and re-save it from there, or best, copy the DreamWeaver file that was distributed to the class in the BERKwork folder and as an attachment to an e-mail message sent by the Instructor.

ADVANCED PICTURE SIZING TIPS

All pictures in the slide show must be the same size. HTML only allows changing the filename, not the size specs. Make all pictures in the slide show sequence the same proportions and the same orientation, otherwise the flow could be uneven.

If both vertical and horizontal sets are to be mixed together however, then all pictures must be made perfectly square by adding padding to each that is the same color as the background of the web page. Set the background color in Photoshop to the same color of the web page background color, then use the IMAGE> CANVAS SIZE command.

add padding on the left side of all right hand vertical pictures

add padding on the right side of all left hand vertical pictures

add padding to the top and bottom of any horizontal pictures

Our target is viewers with at least a 15" laptop, which allows a browser window to be opened to 1440 x 900 pixels. Allowing for some space on either side of the pictures and a little space in between leaves about 600 pixels per picture. In this case, a set of square pictures at 500 x 500 pixels is optimum.