

Dynamic Flash 01

loadVariablesNum()

getURL()

Usability

loadVariablesNum() [E resources/flash_guestbook; dynamic_flash_1]

Syntax:

```
loadVariables("url", level [, variables])
```

Action which reads data from an external file, such as a text file or text generated by a CGI script, Active Server Pages (ASP), or PHP, or Perl script, and sets the values for variables in a Flash Player level. This action can also be used to update variables in the active movie with new values.

e.g.

```
loadVariablesNum("archive.php", 0);
```

getURL()

Loads a document from a specific URL into a window or passes variables to another application at a defined URL.

```
getURL(url [, window [, "variables"]])
```

`url` The URL from which to obtain the document.

`window` An optional parameter specifying the window or HTML frame that the document should load into. You can enter the name of a specific window or choose from the following reserved target names:

`_self` specifies the current frame in the current window.

`_blank` specifies a new window.

`_parent` specifies the parent of the current frame.

`_top` specifies the top-level frame in the current window.

`variables` A GET or POST method for sending variables.

```
on (release) {  
    getURL("addtext.php?name="+name+"&txt="+txt+"");  
}
```

Usability (applies for Flash and Dynamic Web Design)

- Consistency: do not use different words icons if they mean the same thing. Take advantage of your design skills and all parameters in design, such as position, scale, color, behaviour and response to be consistent.
- Define design variables and constants.
- Connect to concepts which are familiar to the user, follow real-world conventions when designing customized metaphors
- Make transparent where users are located within the system, allow users to estimate the potential of the whole system.
- Define an appropriate 'pace' for user perception and interaction. Balance the amount of information and required navigation. Prefer direct links as opposed to back/forward.

Usability

- Empower the user, avoid linear sequences without ‘emergency exits’, don’t trap the user. Anticipate users’ interests/needs and offer a line of least resistance rather than a single path.
- Avoid layering information, make it easy to stay within the system, identify exits.
- Develop structure and interaction model as paper sketches. Develop screen designs in Photoshop or Illustrator. Divide technical problems in simple sub-problems.
- Test your interface, let other people use it. Observe and interpret problems.

Usability

- Being aware of the rules for usability, defining your goal, your target audience, applying your design methodology, taking to heart users' feedback:

Break the rules if appropriate and necessary. Know , when to break them.

Don't lose the big picture by getting obsessed with details or getting carried away with technical possibilities.

What is difficult with Flash?

- large data and database driven sites
- searching, search engines cannot search flash movies
- super-fast loading and performing pages
- handling large amounts of texts
- standard functions such as scrollbars, pupups, forms etc. need to be replicated using the Flash Components.
- Flash Remoting and The AMFPHP (www.amfphp.org) open-source project is a powerful way to combine the advantages of a server-side pre-processor with the advantages of Flash

What is appropriate for Flash?

- interactive/reactive graphics and animations, sound applications
- vector graphics and typography
- customized interfaces
- MX 2004 supports video (proprietary .flv format)