JXA Cheatsheet for macOS Application Scripting and Automation

JXA (JavaScript for Automation) is a scripting language that allows you to automate tasks on your Mac using JavaScript. Here's a quick overview of its unique features and some code blocks for common macOS application scripting and automation tasks.

Terminology

- **Application**: A running instance of a macOS application.
- **Process**: A running instance of an application.
- **Window**: A window in an application.
- **Tab**: A tab in a window.
- **Document**: A document in an application.
- **UI Element**: A user interface element, such as a button, text field, or menu item.

Variables

Declare variables using the `var` keyword. JXA is dynamically typed and variables can hold any type of value.

```
var app = Application("Safari");
var window = app.windows[0];
```

Functions

Functions in JXA are declared using the `function` keyword. They can take parameters and return values.

```
function clickButton(buttonName) {
 var app = Application.currentApplication();
 app.includeStandardAdditions = true;
 app.click(app.button(buttonName));
}
```

Loops

JXA supports `for` and `while` loops for iterating over arrays or performing a task a certain number of times.

```
var app = Application("Safari");
var windows = app.windows;

for (var i = 0; i < windows.length; i++) {
 console.log(windows[i].name());
}

var i = 0;
while (i < windows.length) {
 console.log(windows[i].name());
 i++;
}
```

Conditionals

Use `if` statements to execute code based on a condition. `else if` and `else` statements can be used to handle multiple conditions.

```
var app = Application("Safari");
var window = app.windows[0];

if (window.exists()) {
 console.log("Window exists");
} else {
 console.log("Window does not exist");
}
```

File Manipulation

JXA provides built-in support for file manipulation, including reading and writing files, and interacting with the file system.

```
var file = File("/path/to/file.txt");

// Read the entire contents of the file
var contents = file.read();

// Write new contents to the file
file.write("New contents");

// Check if the file exists
if (file.exists()) {
 console.log("File exists");
}

// Get information about the file
console.log(file.creationDate());
console.log(file.modificationDate());
console.log(file.size());
```

Resources

Here are some resources to help you learn more about JXA for macOS application scripting and automation:

- JXA Cookbook
- JXA Resources
- macOS Automation (Apple Developer Documentation)
- UI Browser (Tool for exploring the user interface of macOS applications)