EXTENDING ADOBE CAPTIVATE WITH JAVASCRIPT

ADVANCED TECHNIQUES FROM A WEB DEVELOPER'S PERSPECTIVE

HTTPS://GITHUB.COM/SDWARWICK/CAPTIVATE-DEMOS

STEVEN WARWICK, ELEARNINGOCEAN.COM
SDWARWICK@ELEARNINGOCEAN.COM

CONTEXT

- Captivate
- HTML projects
- “Responsive” design
- Windows 10 development environment
- JavaScript ECMA 2015
- Chrome browser
- Notepad++ text editor

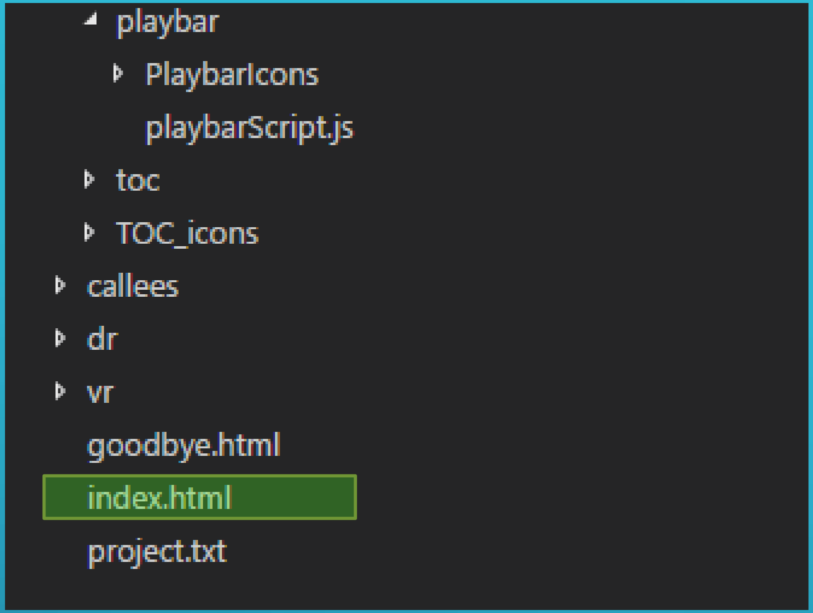
PLAN

- Captivate as a web development platform
- Efficient development of JavaScript/Captivate scripts
- Example Scripts
  - Fully custom quiz interactions
  - Full-screen mode
  - D&D
- Adobe documented vs. undocumented functions
  - Bridging between JavaScript and Captivate
- Overview of other possibilities with JavaScript
- Questions

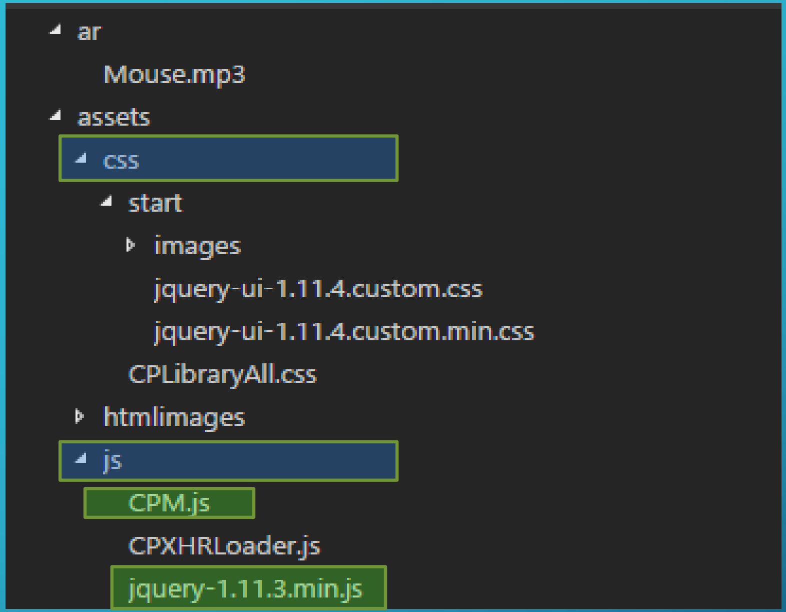
CAPTIVATE FROM THE WEB DEVELOPERS PERSPECTIVE

- WYSIWYG website builders:
    - "Closed" builders generate sites that cannot easily be modified after being generated
        - Easy to get started building, limited access to potential of modern design
        - Weebly, Wix, Squarespace
    - "Open" builders support direct modification of generated sites & continued editing
        - Deeper understanding of web technologies needed
        - Pinegrow, Bootstrap Studio, Bootply
- Captivate – 90% closed / 10% open
- Custom features valuable for eLearning
- Reasonable strategy given initial target audience

ANATOMY OF A WEBSITE (CAPTIVATE FILE LAYOUT)



```
  ◢ playbar
     ▷ PlaybarIcons
       playbarScript.js
  ▷ toc
  ▷ TOC_icons
▷ callees
▷ dr
▷ vr
  goodbye.html
  index.html
  project.txt
```

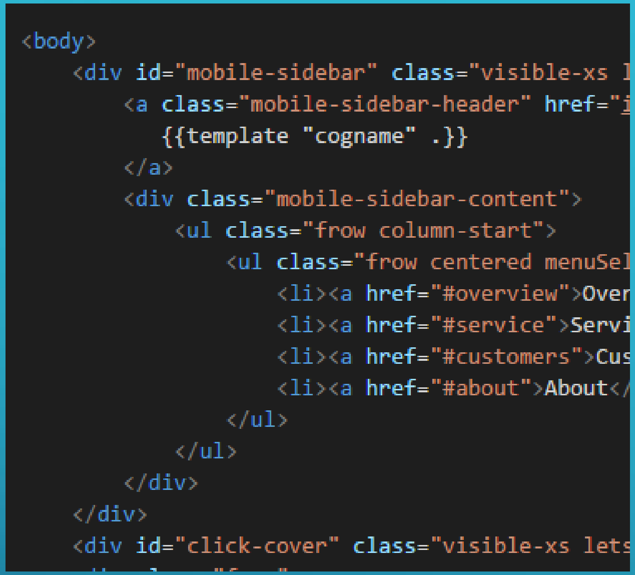


```
◢ ar
    Mouse.mp3
◢ assets
  ◢ css
    ◢ start
      ▷ images
        jquery-ui-1.11.4.custom.css
        jquery-ui-1.11.4.custom.min.css
      CPLibraryAll.css
  ▷ htmlimages
  ◢ js
      CPM.js
      CPXHRLoader.js
      jquery-1.11.3.min.js
```

- A module produced by Captivate is structured in a very common website design style
- A zipped module is simply a single-file version of this exact directory structure
- When a captivate module is loaded into an LMS, the zip file is simply uncompressed by the LMS
- Websites typically need to be “served” by a server program (apache/nginx) in case external content needs to be loaded
- When all content is inside the module directory, a browser can be used to view the website (file://)

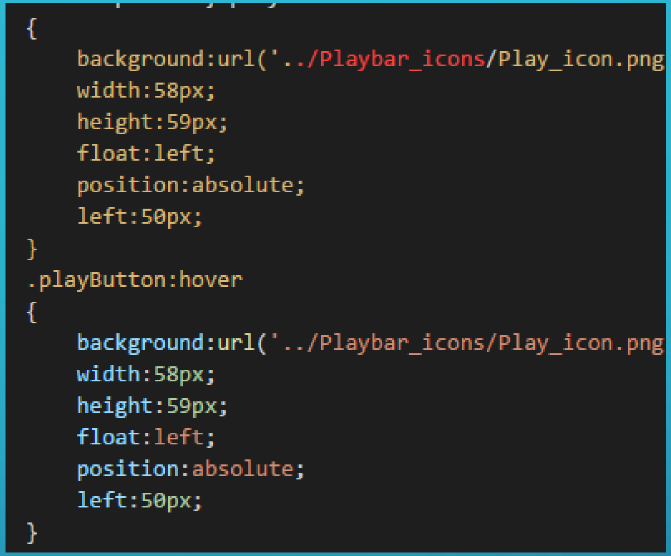
ANATOMY OF A CAPTIVATE WEBSITE

HTML



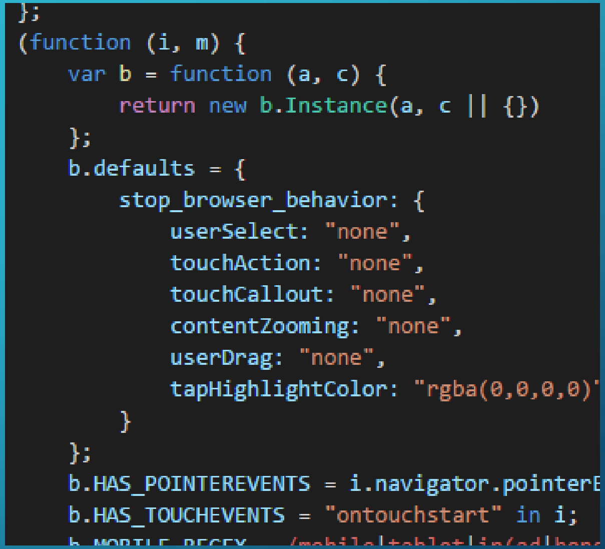
```
<body>
    <div id="mobile-sidebar" class="visible-xs l
        <a class="mobile-sidebar-header" href="i
           {{template "cogname" .}}
        </a>
        <div class="mobile-sidebar-content">
            <ul class="frow column-start">
                <ul class="frow centered menuSel
                    <li><a href="#overview">Over
                    <li><a href="#service">Servi
                    <li><a href="#customers">Cus
                    <li><a href="#about">About</
                </ul>
            </ul>
        </div>
    </div>
    <div id="click-cover" class="visible-xs lets
```

CSS



```
{
    background:url('../Playbar_icons/Play_icon.png
    width:58px;
    height:59px;
    float:left;
    position:absolute;
    left:50px;
}
.playButton:hover
{
    background:url('../Playbar_icons/Play_icon.png
    width:58px;
    height:59px;
    float:left;
    position:absolute;
    left:50px;
}
```

Javascript



```
};
(function (i, m) {
    var b = function (a, c) {
        return new b.Instance(a, c || {})
    };
    b.defaults = {
        stop_browser_behavior: {
            userSelect: "none",
            touchAction: "none",
            touchCallout: "none",
            contentZooming: "none",
            userDrag: "none",
            tapHighlightColor: "rgba(0,0,0,0)"
        }
    };
    b.HAS_POINTEREVENTS = i.navigator.pointerE
    b.HAS_TOUCHEVENTS = "ontouchstart" in i;
    b.MOBILE_REGEX = /mobile|tablet|ip(ad|hon
```

- Same structures are seen in Captivate as in all websites
- "CPM.js" file contains
  - All content data – shapes, text, timing, placement, quiz
  - Captivate JavaScript Library that "runs" the website
  - Since the file is compressed, it is hard to decipher

WHY JAVASCRIPT?

- Most popular programming language – StackOverflow / Github
- Used for both user interaction in browser and business logic on server
  - Access all the power of the browser
- Completely free development environment
- All Browsers have powerful, built-in debugging tools
- Very fast design/test cycle - no "publishing/compiling" process
- Knowledge on demand - profound change in learning process
  - Stackoverflow http://stackoverflow.com/insights/survey/2016
  - 2.7Million questions, 3.2Million answers in 2015
  - Thousands of tutorials

WHY USE JAVASCRIPT WITH CAPTIVATE

Upside

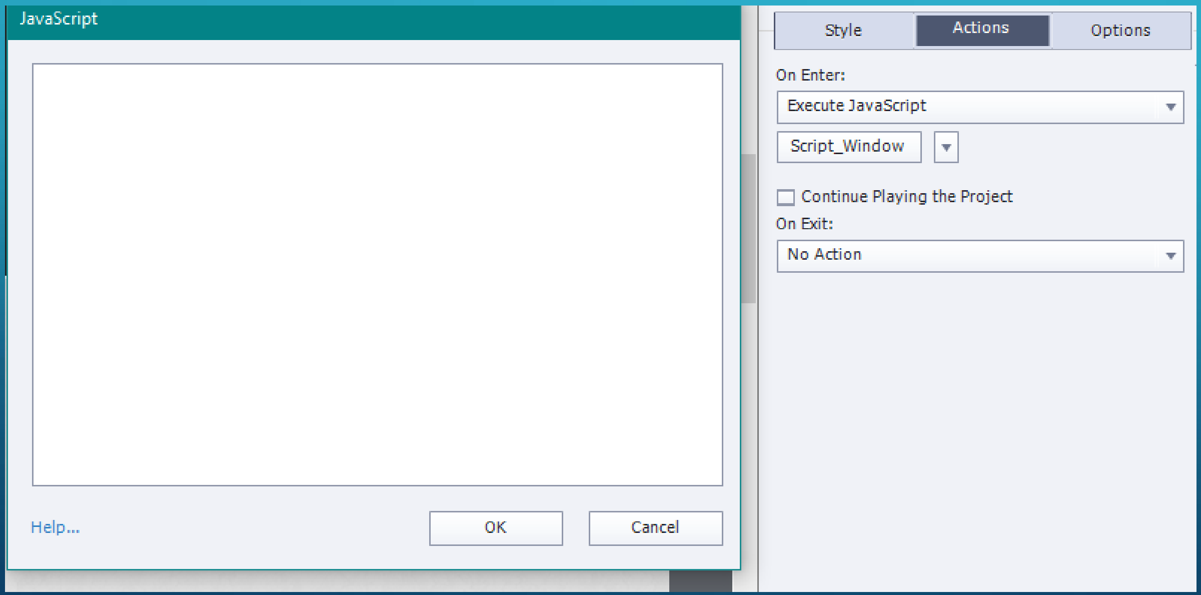
- All "Automation" functions in one place – Model/View/Controller Architecture
- JavaScript can control any aspect of UI
  - Change shape properties, display notifications, react to any user event
  - Create custom quiz interactions, unique animations etc..
- JavaScript functions can be debugged while the presentation is running, unlike advanced actions
- Many online tutorials for using JavaScript with Captivate
  - Large subject area, no tutorial is can be comprehensive – point solutions and examples

Downside

- Steeper learning curve – HTML/CSS/Jquery/Captivate
- Lots of cool stuff is undocumented, discovered and published by developers

HOW TO WORK EFFICIENTLY WITH JAVASCRIPT

- Internally supported approach: Use built-in JavaScript script window
  - No syntax checking
  - Must re-publish module to update
  - Hard to maintain, code is sprinkled throughout the modules

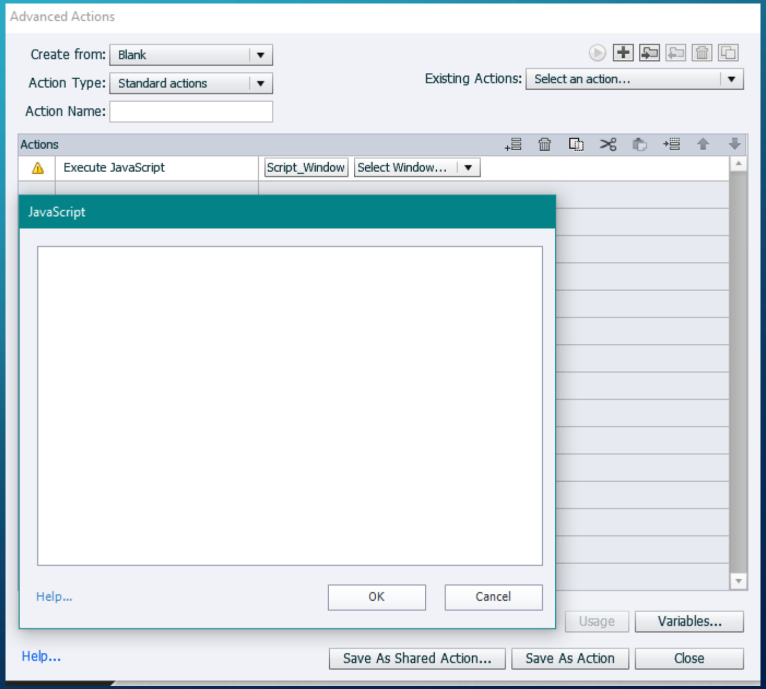


On Enter: Execute JavaScript

Script_Window

□ Continue Playing the Project

On Exit: No Action



Create from: Blank

Action Type: Standard actions

Action Name:

Existing Actions: Select an action...

Execute JavaScript | Script_Window | Select Window...

HOW TO WORK EFFICIENTLY WITH JAVASCRIPT

Better approach:

- External file holds all JavaScript functions
- Changes in file will be loaded whenever the module is viewed, no need to re-publish course – rapid development!

Downside:

- Files “outside” a module are only accessible when using http:// not file://
  - No Captivate “preview” mode - must “publish”
  - Use local web server
  - Move file inside module - automation

HOW TO WORK EFFICIENTLY WITH JAVASCRIPT

On enter execute JavaScript + continue playing project

```
if( !externLoaded ) {
$('body').append('<script src="../multichoice.js" async="false"></script>');

$(fontLink).appendTo("head");

externLoaded = true;
}
```

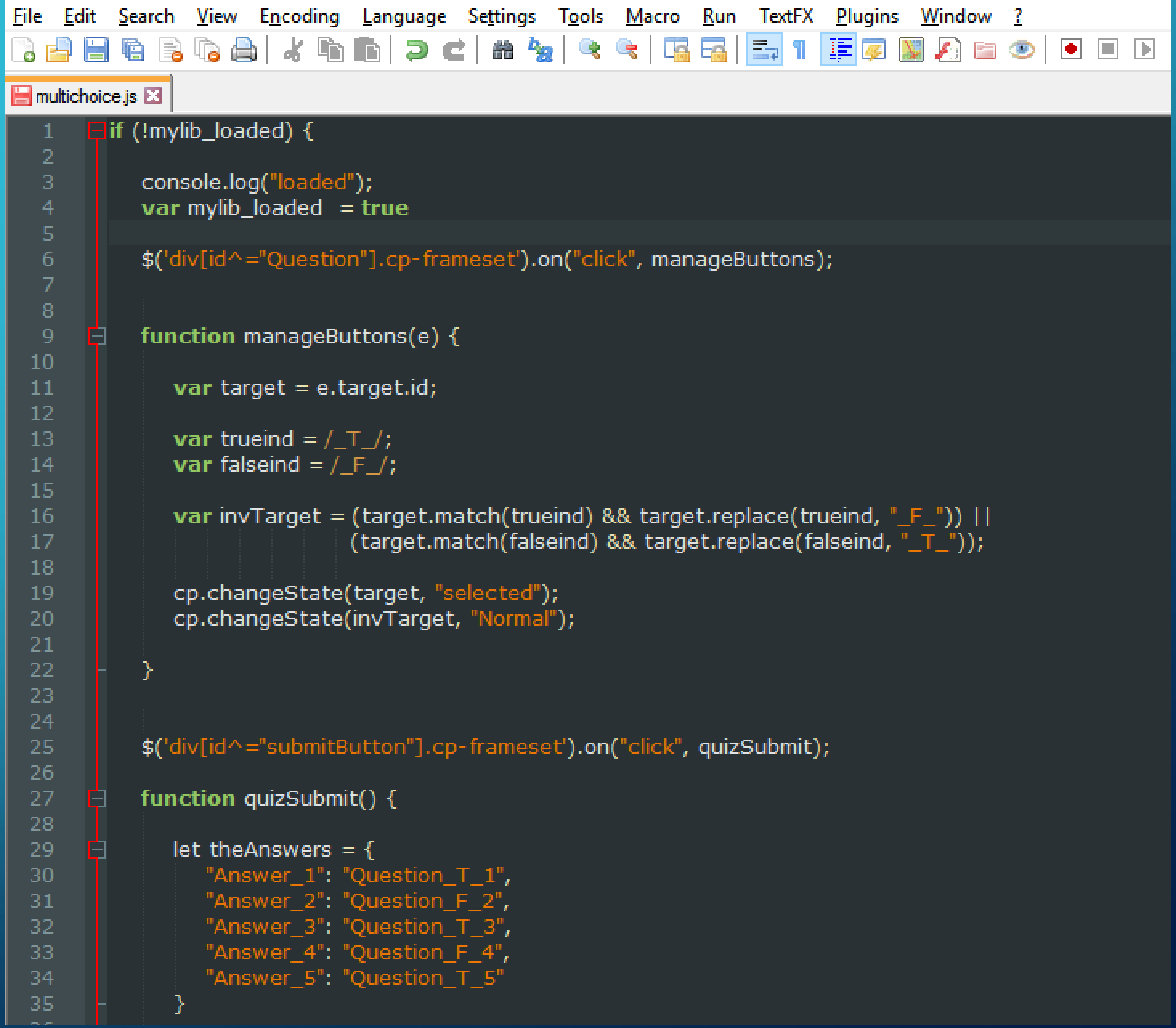
- JavaScript file is outside of course module, is not deleted when module is re-published
- Add to every slide in cases where LMS can jump past first slide

HOW TO WORK EFFICIENTLY WITH JAVASCRIPT

Notepad++ text editor as example

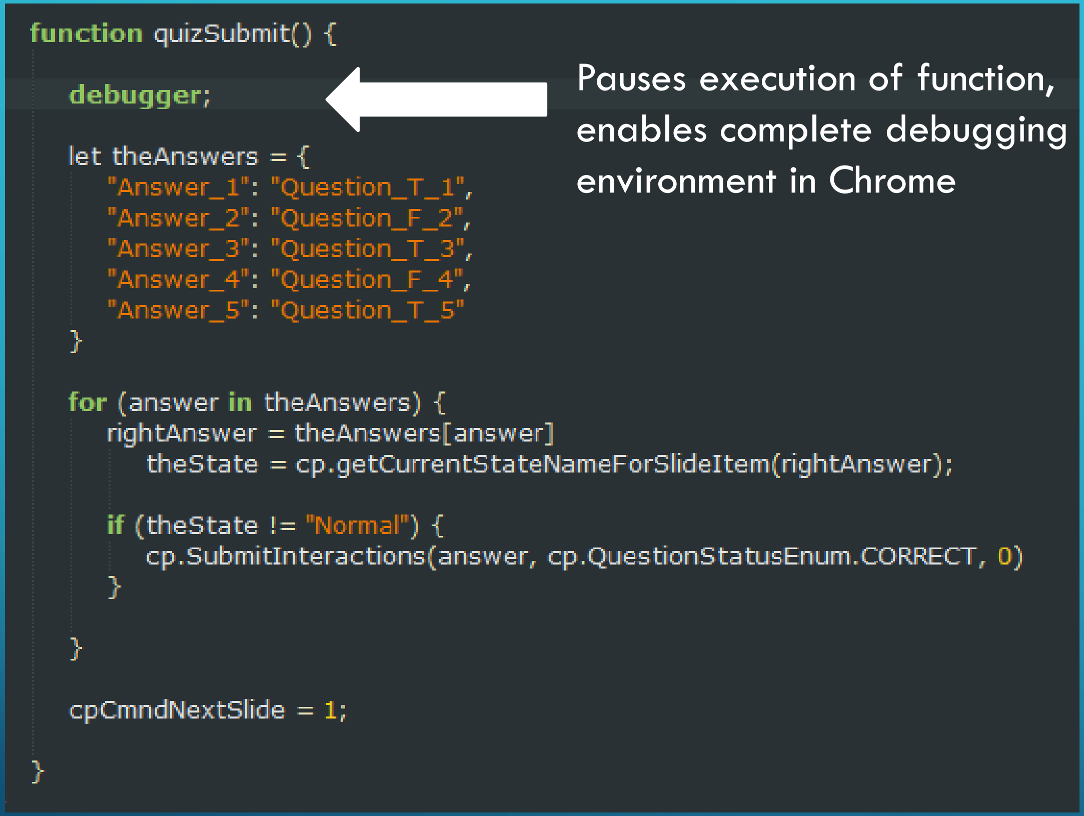
Far easier than built-in script window!

- JavaScript syntax and error highlighting
- Variable name validation
- Multiple windows, spell check etc.



```
if (!mylib_loaded) {

    console.log("loaded");
    var mylib_loaded  = true

    $('div[id^="Question"].cp-frameset').on("click", manageButtons);


    function manageButtons(e) {

        var target = e.target.id;

        var trueind = /_T_/;
        var falseind = /_F_/;

        var invTarget = (target.match(trueind) && target.replace(trueind, "_F_")) ||
                        (target.match(falseind) && target.replace(falseind, "_T_"));

        cp.changeState(target, "selected");
        cp.changeState(invTarget, "Normal");

    }


    $('div[id^="submitButton"].cp-frameset').on("click", quizSubmit);

    function quizSubmit() {

        let theAnswers = {
            "Answer_1": "Question_T_1",
            "Answer_2": "Question_F_2",
            "Answer_3": "Question_T_3",
            "Answer_4": "Question_F_4",
            "Answer_5": "Question_T_5"
        }
```

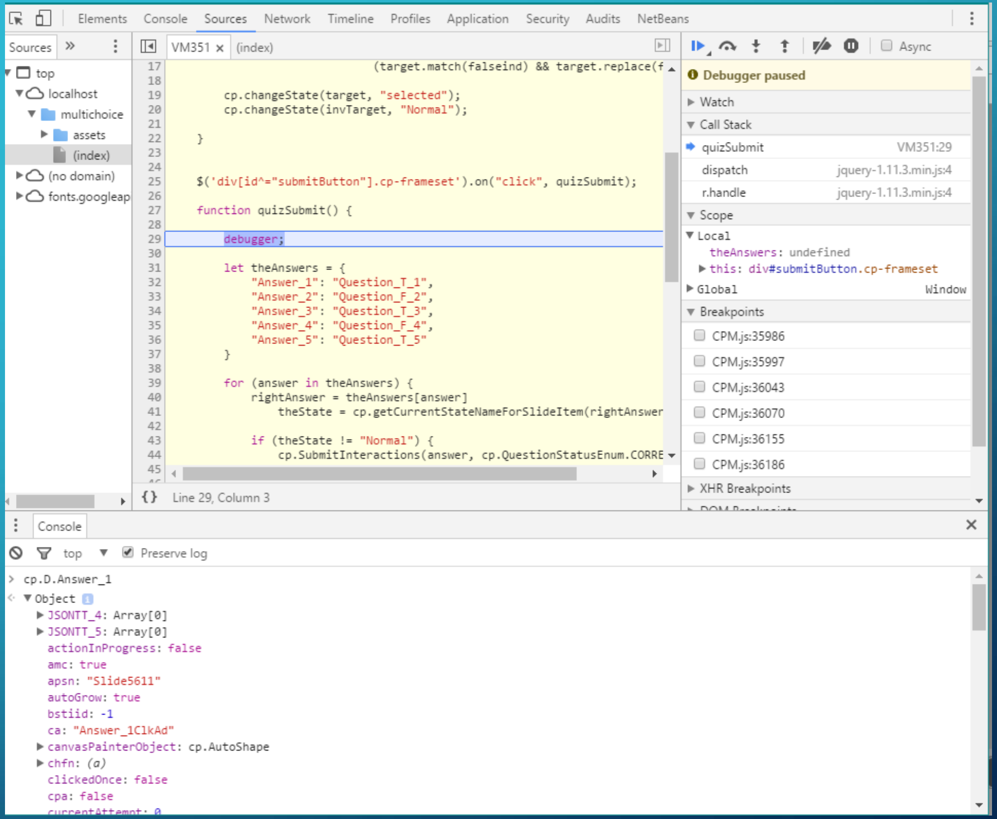
DEBUGGING JAVASCRIPT WITH CHROME



```
function quizSubmit() {

    debugger;

    let theAnswers = {
        "Answer_1": "Question_T_1",
        "Answer_2": "Question_F_2",
        "Answer_3": "Question_T_3",
        "Answer_4": "Question_F_4",
        "Answer_5": "Question_T_5"
    }

    for (answer in theAnswers) {
        rightAnswer = theAnswers[answer]
            theState = cp.getCurrentStateNameForSlideItem(rightAnswer);

        if (theState != "Normal") {
            cp.SubmitInteractions(answer, cp.QuestionStatusEnum.CORRECT, 0)
        }

    }

    cpCmndNextSlide = 1;

}
```

Pauses execution of function, enables complete debugging environment in Chrome

F12 opens Chrome debugger!



```
                                    (target.match(falseind) && target.replace(f

        cp.changeState(target, "selected");
        cp.changeState(invTarget, "Normal");

    }


    $('div[id^="submitButton"].cp-frameset').on("click", quizSubmit);

    function quizSubmit() {

        debugger;

        let theAnswers = {
            "Answer_1": "Question_T_1",
            "Answer_2": "Question_F_2",
            "Answer_3": "Question_T_3",
            "Answer_4": "Question_F_4",
            "Answer_5": "Question_T_5"
        }

        for (answer in theAnswers) {
            rightAnswer = theAnswers[answer]
                theState = cp.getCurrentStateNameForSlideItem(rightAnswer

            if (theState != "Normal") {
                cp.SubmitInteractions(answer, cp.QuestionStatusEnum.CORRE
```

Line 29, Column 3

Debugger paused

quizSubmit	VM351:29
dispatch	jquery-1.11.3.min.js:4
r.handle	jquery-1.11.3.min.js:4

```
▼ Local
    theAnswers: undefined
  ▶ this: div#submitButton.cp-frameset
▶ Global                              Window
```

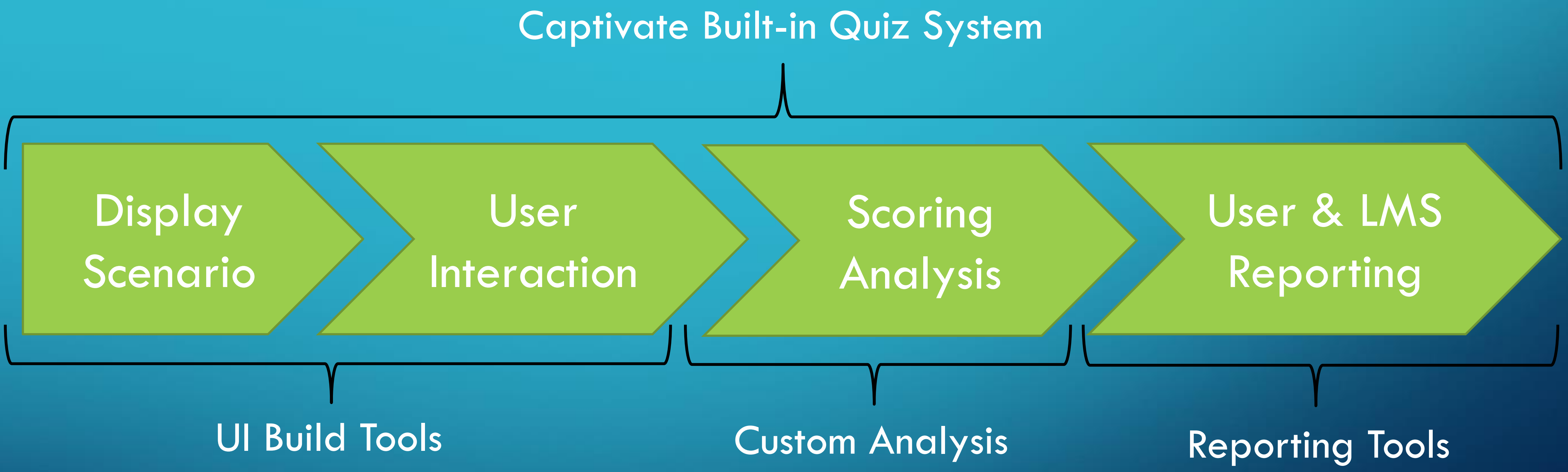
- □ CPM.js:35986
- □ CPM.js:35997
- □ CPM.js:36043
- □ CPM.js:36070
- □ CPM.js:36155
- □ CPM.js:36186

```
> cp.D.Answer_1
< ▼ Object
    ▶ JSONTT_4: Array[0]
    ▶ JSONTT_5: Array[0]
      actionInProgress: false
      amc: true
      apsn: "Slide5611"
      autoGrow: true
      bstiid: -1
      ca: "Answer_1ClkAd"
    ▶ canvasPainterObject: cp.AutoShape
    ▶ chfn: (a)
      clickedOnce: false
      cpa: false
      currentAttempt: 0
```

Step-by-step debugging – unlike advanced actions

CUSTOM QUIZ INTERACTION

HTTPS://GITHUB.COM/SDWARWICK/CAPTIVATE-DEMOS

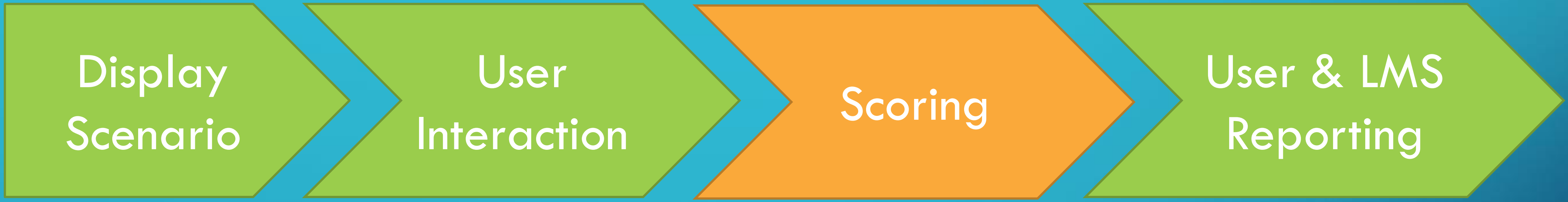


Fully customized quiz interactions

Connect interaction to scoring to reporting

CUSTOM QUIZ INTERACTION

HTTPS://GITHUB.COM/SDWARWICK/CAPTIVATE-DEMOS



Display Scenario

User Interaction

Scoring

User & LMS Reporting

Requires connecting into Captivate library at undocumented points

Three strategies to create connection between interaction, scoring and reporting

- Hidden scored items, activated by JavaScript
- Modifying database, changing score value of an item
- Replacing the entire scoring function

EXAMPLE – CUSTOM QUIZ INTERACTION

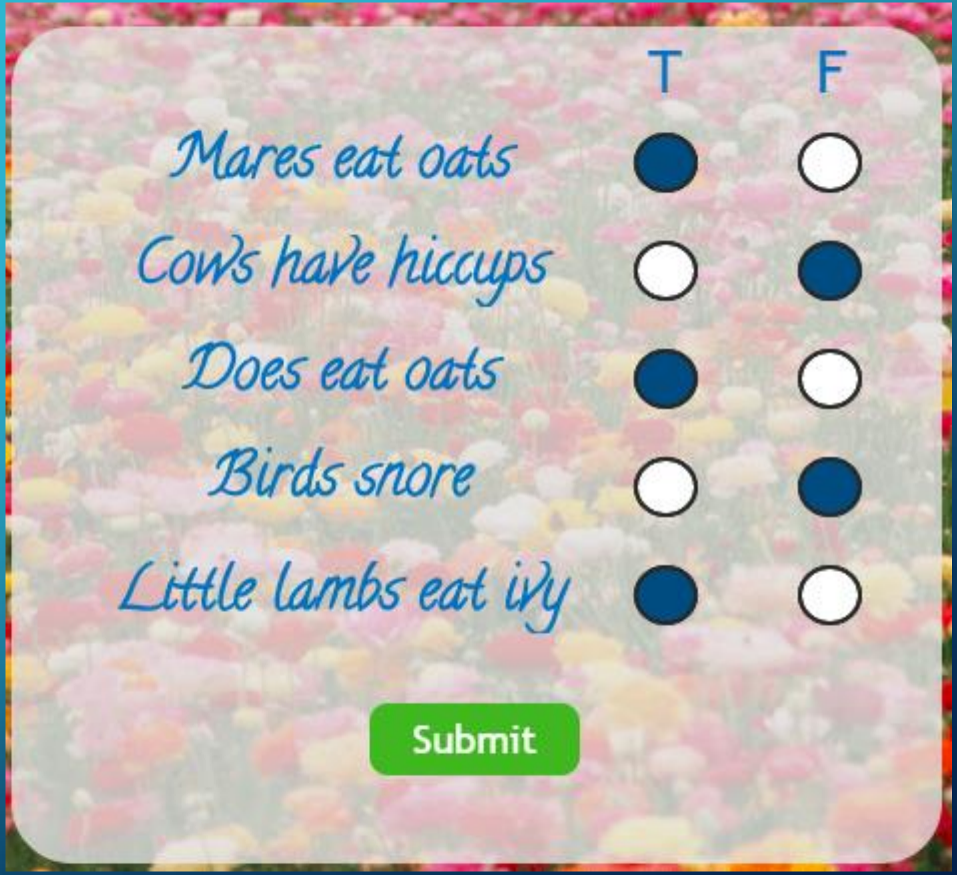
HTTPS://GITHUB.COM/SDWARWICK/CAPTIVATE-DEMOS

Rules:

- No scoring until “Submit” is pressed
- True/false toggles correctly
- Score for each answer may be different
- +25 points for 4/5 right answers
- +50 points for 5/5 right answers

Strategy:

- All user interactions managed by JavaScript
- Hidden scored items, activated by JavaScript



	T	F
Mares eat oats	●	○
Cows have hiccups	○	●
Does eat oats	●	○
Birds snore	○	●
Little lambs eat ivy	●	○

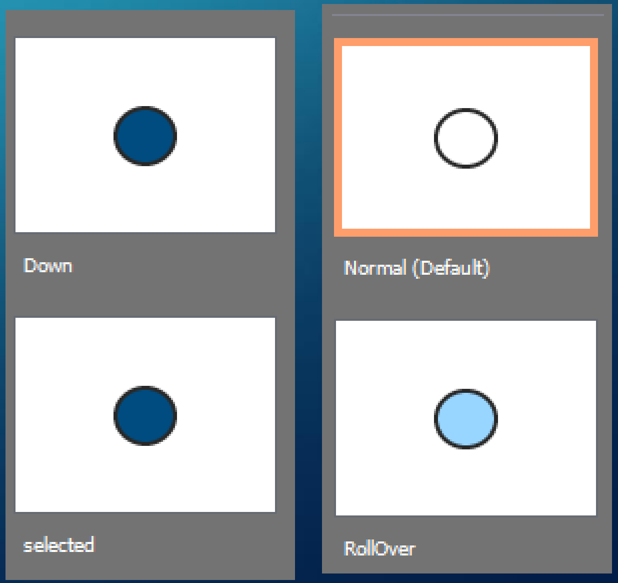
EXAMPLE – CUSTOM QUIZ INTERACTION

- Slide “on enter execute JavaScript”: Add script file and links to fonts

```
$('body').append('<script src="../multichoice.js" async="false"></script>');

var fontLink = '<link href="https://fonts.googleapis.com/css?family=Calligraffitti" rel="stylesheet">';
$(fontLink).appendTo("head");
```

- All buttons are simple circle smartshapes with “use as button”

- Create objec state called “selected”
- Controlled by JavaScript



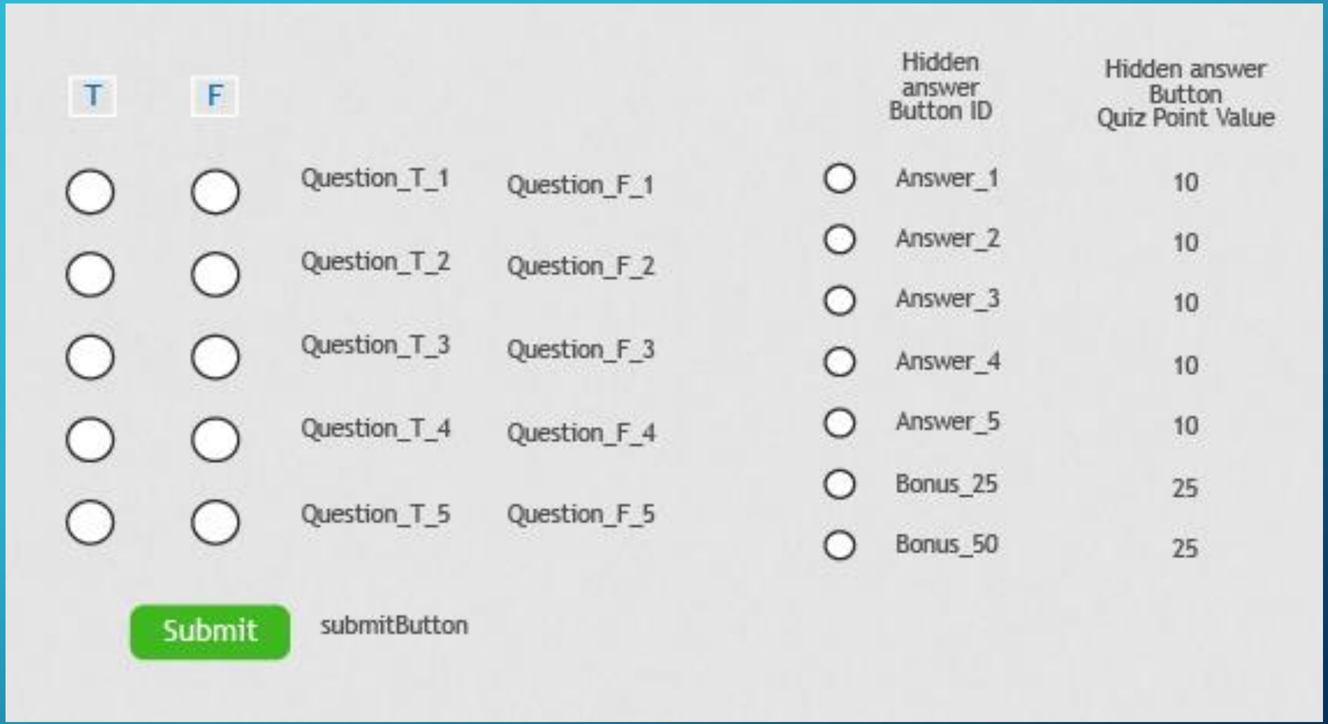
EXAMPLE – CUSTOM QUIZ INTERACTION

- The shapes are labeled using a regular pattern that will be easily distinguished in the JavaScript Code

- The hidden answer buttons are all set to “Include in Quiz” and points can be assigned to each answer

- Add variables to enable connection between JavaScript and Captivate

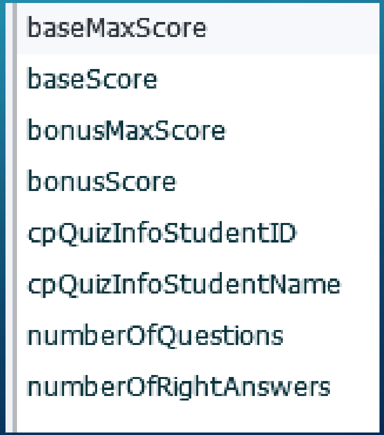
- That’s it.. no advanced actions



T	F		
○	○	Question_T_1	Question_F_1
○	○	Question_T_2	Question_F_2
○	○	Question_T_3	Question_F_3
○	○	Question_T_4	Question_F_4
○	○	Question_T_5	Question_F_5

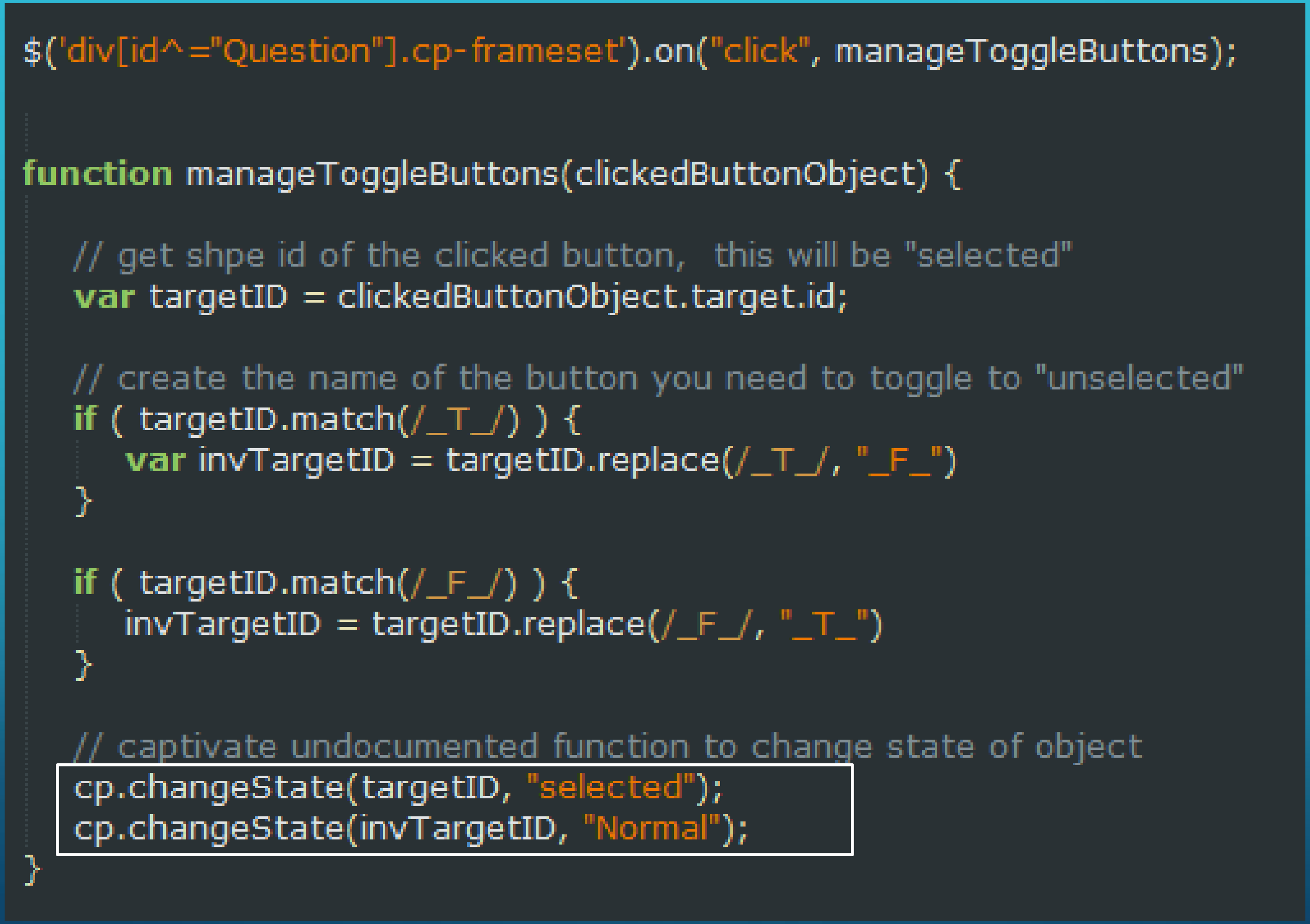
	Hidden answer Button ID	Hidden answer Button Quiz Point Value
○	Answer_1	10
○	Answer_2	10
○	Answer_3	10
○	Answer_4	10
○	Answer_5	10
○	Bonus_25	25
○	Bonus_50	25

Submit submitButton



baseMaxScore
baseScore
bonusMaxScore
bonusScore
cpQuizInfoStudentID
cpQuizInfoStudentName
numberOfQuestions
numberOfRightAnswers

EXAMPLE CUSTOM QUIZ INTERACTION - TOGGLE



```
$('div[id^="Question"].cp-frameset').on("click", manageToggleButtons);

function manageToggleButtons(clickedButtonObject) {

    // get shpe id of the clicked button,  this will be "selected"
    var targetID = clickedButtonObject.target.id;

    // create the name of the button you need to toggle to "unselected"
    if ( targetID.match(/_T_/) ) {
        var invTargetID = targetID.replace(/_T_/, "_F_")
    }

    if ( targetID.match(/_F_/) ) {
        invTargetID = targetID.replace(/_F_/, "_T_")
    }

    // captivate undocumented function to change state of object
    cp.changeState(targetID, "selected");
    cp.changeState(invTargetID, "Normal");
}
```

Find all buttons that start with the word “Question”.   When clicked, call “manageToggleButtons function

Take the name of the button that was pressed, changes any “_T_” to “_F_” and any “_F_” to “_T_”

Call an undocumented captivate function “cp.changeState” to toggle between the “Normal” view and the “selected” view

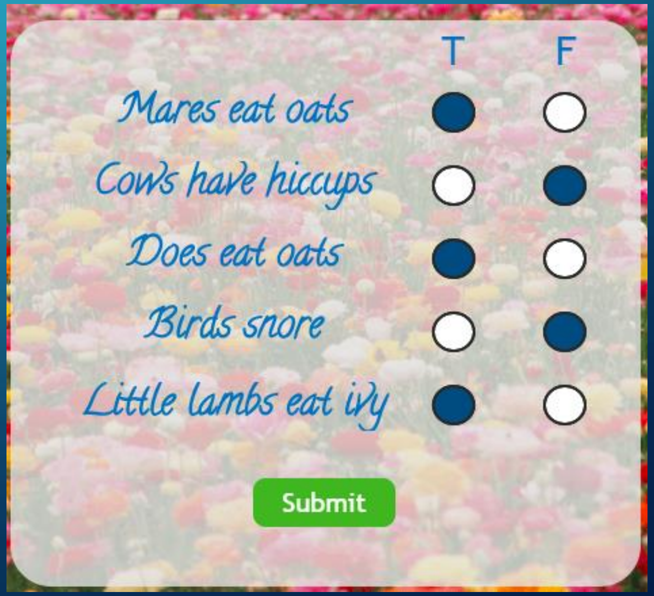
Over the years, many people have contributed to weeding through the CPM.js code to find these functions

EXAMPLE CUSTOM QUIZ INTERACTION - SCORING

1. Did the right buttons get pressed? (state of button tells you)
   - If so, notify captivate by “pressing” a hidden “success” button for each one
2. Did they get 4/5 or 5/5?
   - If so, notify captivate by “pressing” other hidden “success” buttons

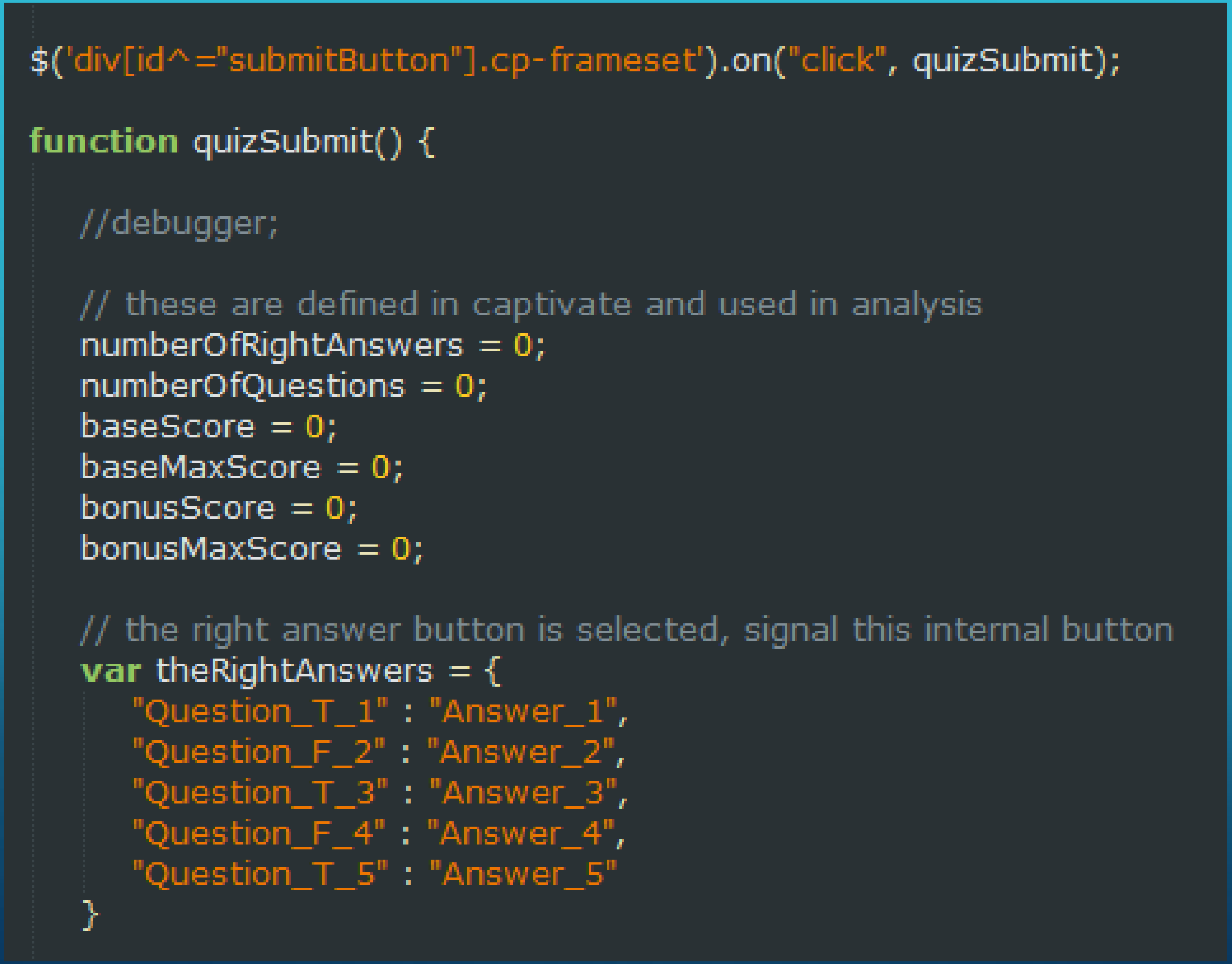
Keep track of totals and report back to captivate:

- Number of questions
- Number of successful answers
- Total possible points
- Actual points scored



	T	F
Mares eat oats	●	○
Cows have hiccups	○	●
Does eat oats	●	○
Birds snore	○	●
Little lambs eat ivy	●	○

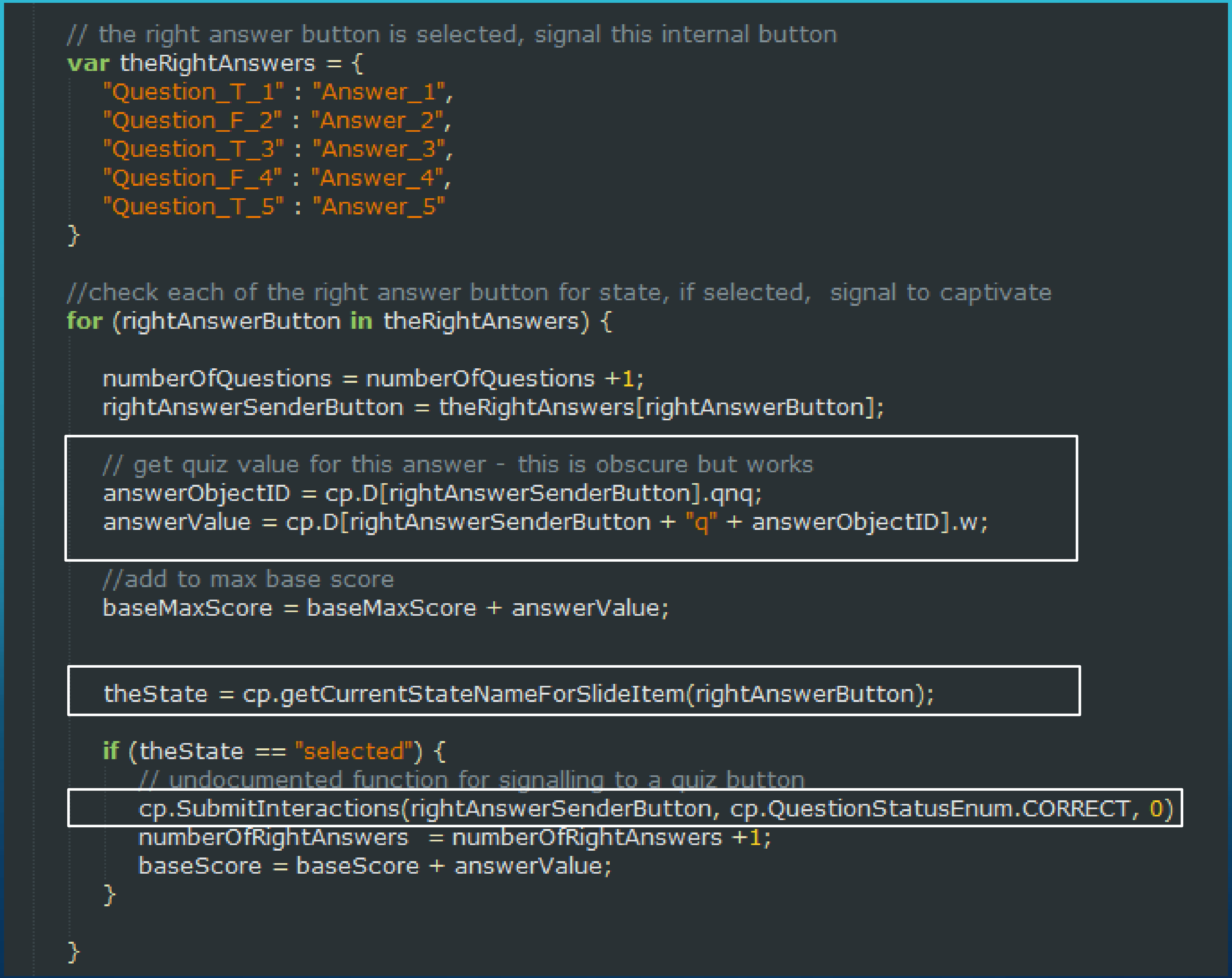
EXAMPLE CUSTOM QUIZ INTERACTION - SCORING



```
$('div[id^="submitButton"].cp-frameset').on("click", quizSubmit);

function quizSubmit() {

    //debugger;

    // these are defined in captivate and used in analysis
    numberOfRightAnswers = 0;
    numberOfQuestions = 0;
    baseScore = 0;
    baseMaxScore = 0;
    bonusScore = 0;
    bonusMaxScore = 0;

    // the right answer button is selected, signal this internal button
    var theRightAnswers = {
        "Question_T_1" : "Answer_1",
        "Question_F_2" : "Answer_2",
        "Question_T_3" : "Answer_3",
        "Question_F_4" : "Answer_4",
        "Question_T_5" : "Answer_5"
    }
```

- The first line triggers the quiz submit function for the button with the ID “submitButton”
- Variables defined in captivate can be directly used in JavaScript!
- The correct answers are defined by which of the question buttons were set to state “selected”
- If the correct answer is selected, which hidden button should be activated?

EXAMPLE CUSTOM QUIZ INTERACTION - SCORING

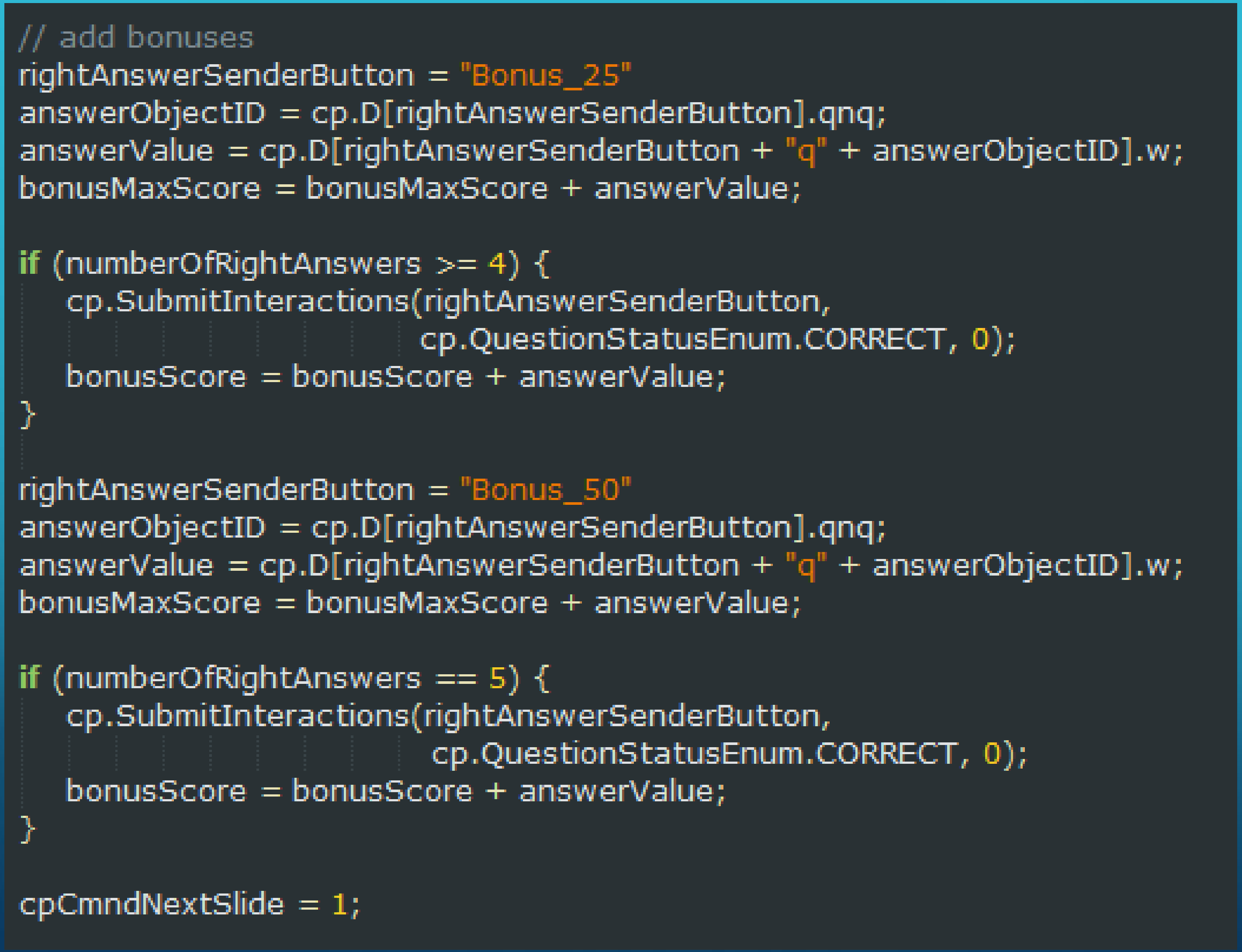


```
// the right answer button is selected, signal this internal button
var theRightAnswers = {
   "Question_T_1" : "Answer_1",
   "Question_F_2" : "Answer_2",
   "Question_T_3" : "Answer_3",
   "Question_F_4" : "Answer_4",
   "Question_T_5" : "Answer_5"
}

//check each of the right answer button for state, if selected,  signal to captivate
for (rightAnswerButton in theRightAnswers) {

   numberOfQuestions = numberOfQuestions +1;
   rightAnswerSenderButton = theRightAnswers[rightAnswerButton];

   // get quiz value for this answer - this is obscure but works
   answerObjectID = cp.D[rightAnswerSenderButton].qnq;
   answerValue = cp.D[rightAnswerSenderButton + "q" + answerObjectID].w;

   //add to max base score
   baseMaxScore = baseMaxScore + answerValue;


   theState = cp.getCurrentStateNameForSlideItem(rightAnswerButton);

   if (theState == "selected") {
      // undocumented function for signalling to a quiz button
      cp.SubmitInteractions(rightAnswerSenderButton, cp.QuestionStatusEnum.CORRECT, 0)
      numberOfRightAnswers  = numberOfRightAnswers +1;
      baseScore = baseScore + answerValue;
   }

}
```

When writing code, try to keep things flexible..

- Determine maximum number of questions, maximum score, answered questions and score values on the fly
- Here’s how to get the value of a quiz button
- Here’s how to find the state of a slide object
- If the right button was selected then we call another undocumented function that signals to captivate that an answer was given correctly.

EXAMPLE CUSTOM QUIZ INTERACTION - SCORING



```
// add bonuses
rightAnswerSenderButton = "Bonus_25"
answerObjectID = cp.D[rightAnswerSenderButton].qnq;
answerValue = cp.D[rightAnswerSenderButton + "q" + answerObjectID].w;
bonusMaxScore = bonusMaxScore + answerValue;

if (numberOfRightAnswers >= 4) {
    cp.SubmitInteractions(rightAnswerSenderButton,
                              cp.QuestionStatusEnum.CORRECT, 0);
    bonusScore = bonusScore + answerValue;
}

rightAnswerSenderButton = "Bonus_50"
answerObjectID = cp.D[rightAnswerSenderButton].qnq;
answerValue = cp.D[rightAnswerSenderButton + "q" + answerObjectID].w;
bonusMaxScore = bonusMaxScore + answerValue;

if (numberOfRightAnswers == 5) {
    cp.SubmitInteractions(rightAnswerSenderButton,
                               cp.QuestionStatusEnum.CORRECT, 0);
    bonusScore = bonusScore + answerValue;
}

cpCmndNextSlide = 1;
```

Find quiz value for the bonus points by looking at the Captivate data

Award points based on some criteria - here it is at least 4 answers right

Here it is 5 answers right…

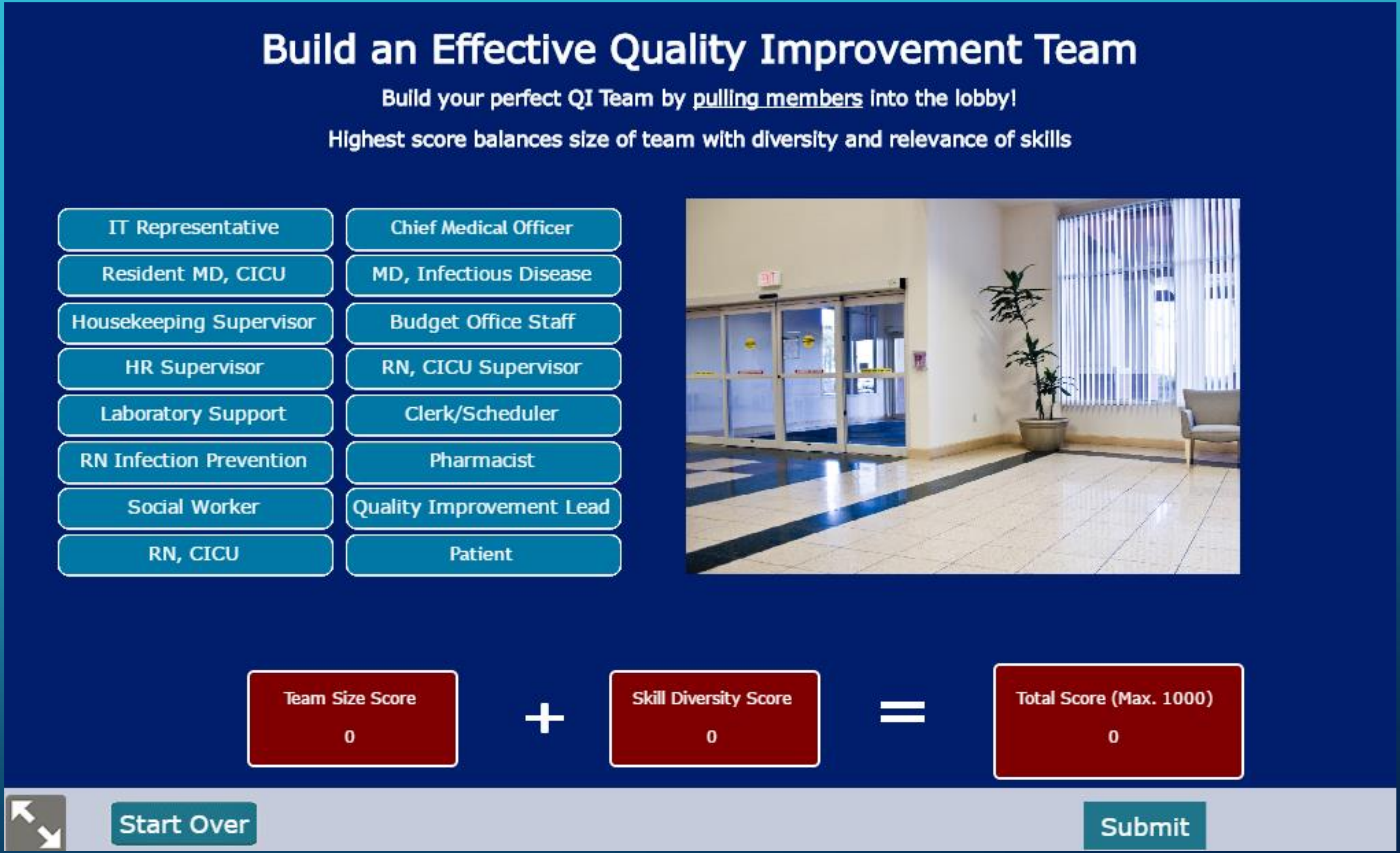
After done, signal to move to next slide by simply setting the “next slide” flag variable

EXAMPLE – CUSTOM QUIZ INTERACTION
WHY IS THIS EXAMPLE IMPORTANT?

- The scoring is completely general
- No advanced actions
- Regular slide, not quiz slide
- Custom interactions can span multiple slides/views
- Other measures can be made along the way:
  - How many times has the user changed their score?
  - How long did it take before the user completed the quiz?
- Scoring doesn't happen for any of the questions until the interaction is complete

EXAMPLE – DRAG AND DROP

HTTPS://GITHUB.COM/SDWARWICK/CAPTIVATE-DEMOS



Build an Effective Quality Improvement Team

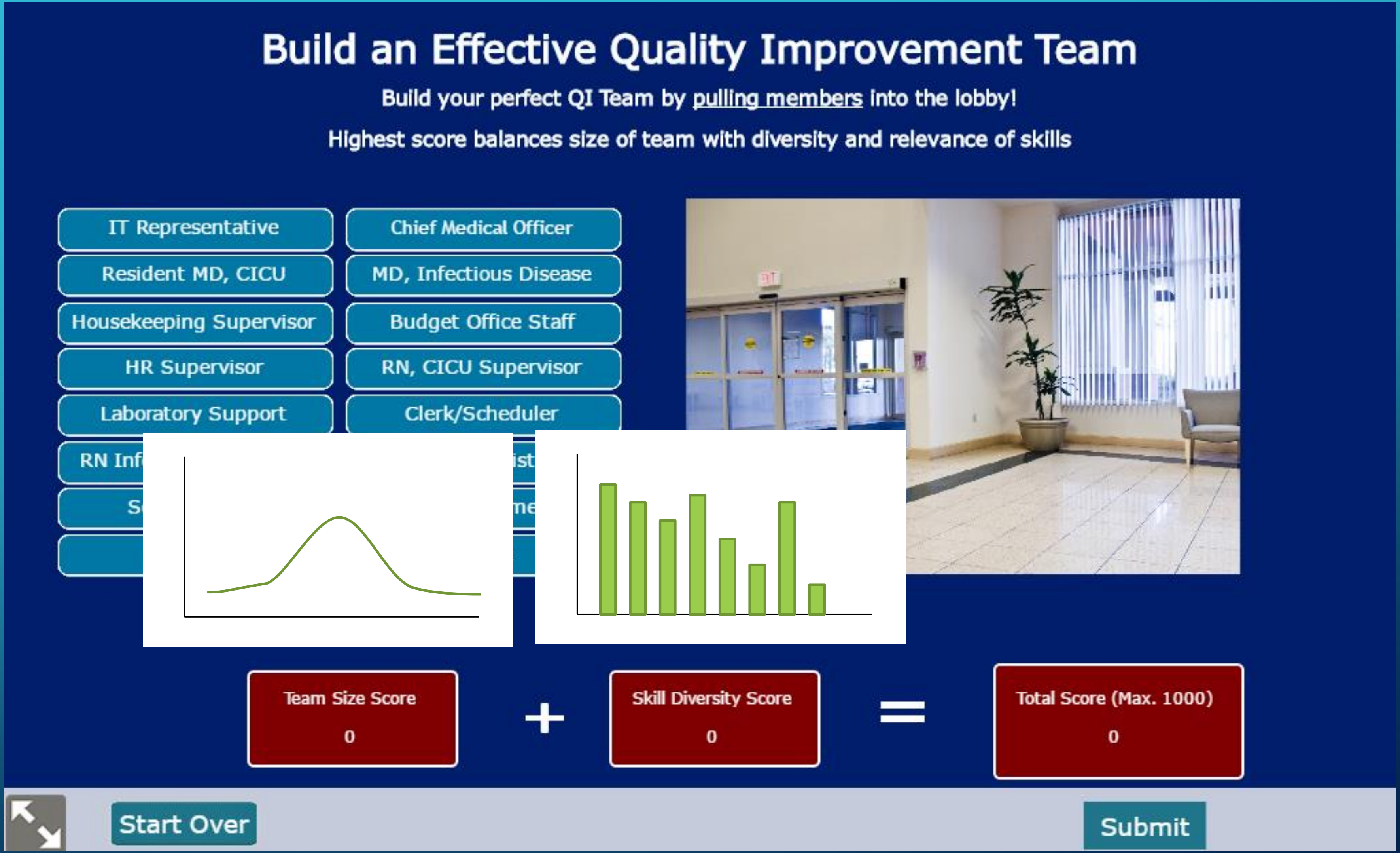
Build your perfect QI Team by pulling members into the lobby!

Highest score balances size of team with diversity and relevance of skills

Team Size Score 0 + Skill Diversity Score 0 = Total Score (Max. 1000) 0

EXAMPLE – DRAG AND DROP CUSTOM SCORING

HTTPS://GITHUB.COM/SDWARWICK/CAPTIVATE-DEMOS



Build an Effective Quality Improvement Team

Build your perfect QI Team by pulling members into the lobby!

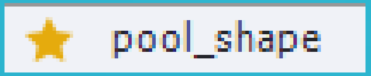
Highest score balances size of team with diversity and relevance of skills

Team Size Score 0 + Skill Diversity Score 0 = Total Score (Max. 1000) 0

EXAMPLE – DRAG AND DROP CUSTOM SCORING

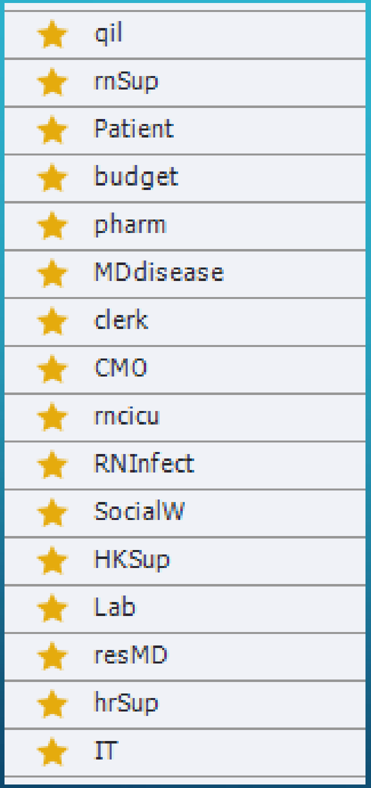
HTTPS://GITHUB.COM/SDWARWICK/CAPTIVATE-DEMOS

"team"
Drop Target



★ pool_shape

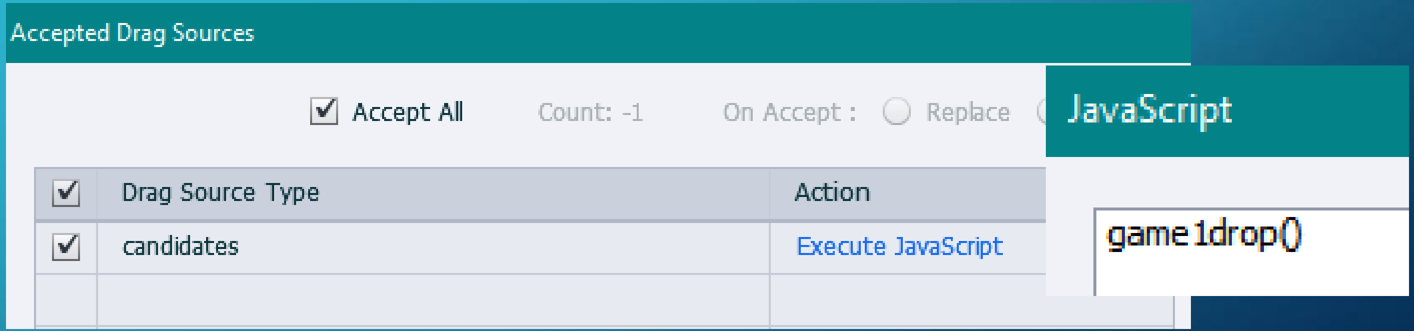
"candidates"
Source Pool



- ★ qil
- ★ rnSup
- ★ Patient
- ★ budget
- ★ pharm
- ★ MDdisease
- ★ clerk
- ★ CMO
- ★ rncicu
- ★ RNInfect
- ★ SocialW
- ★ HKSup
- ★ Lab
- ★ resMD
- ★ hrSup
- ★ IT

All scoring functions in JavaScript
Each time a "candidate" is dropped, "game1drop()" is called

"team"
Object Actions



Accepted Drag Sources

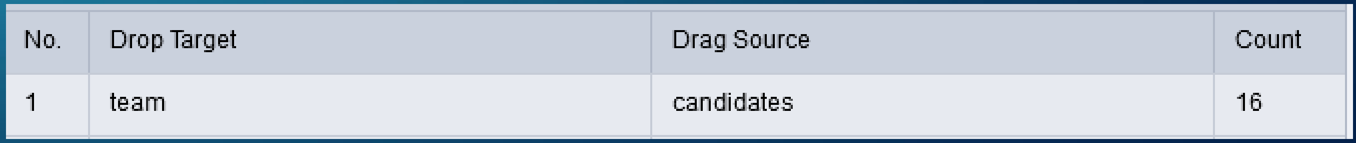
☑ Accept All   Count: -1   On Accept : ○ Replace ○

☑	Drag Source Type	Action
☑	candidates	Execute JavaScript

JavaScript

game1drop()

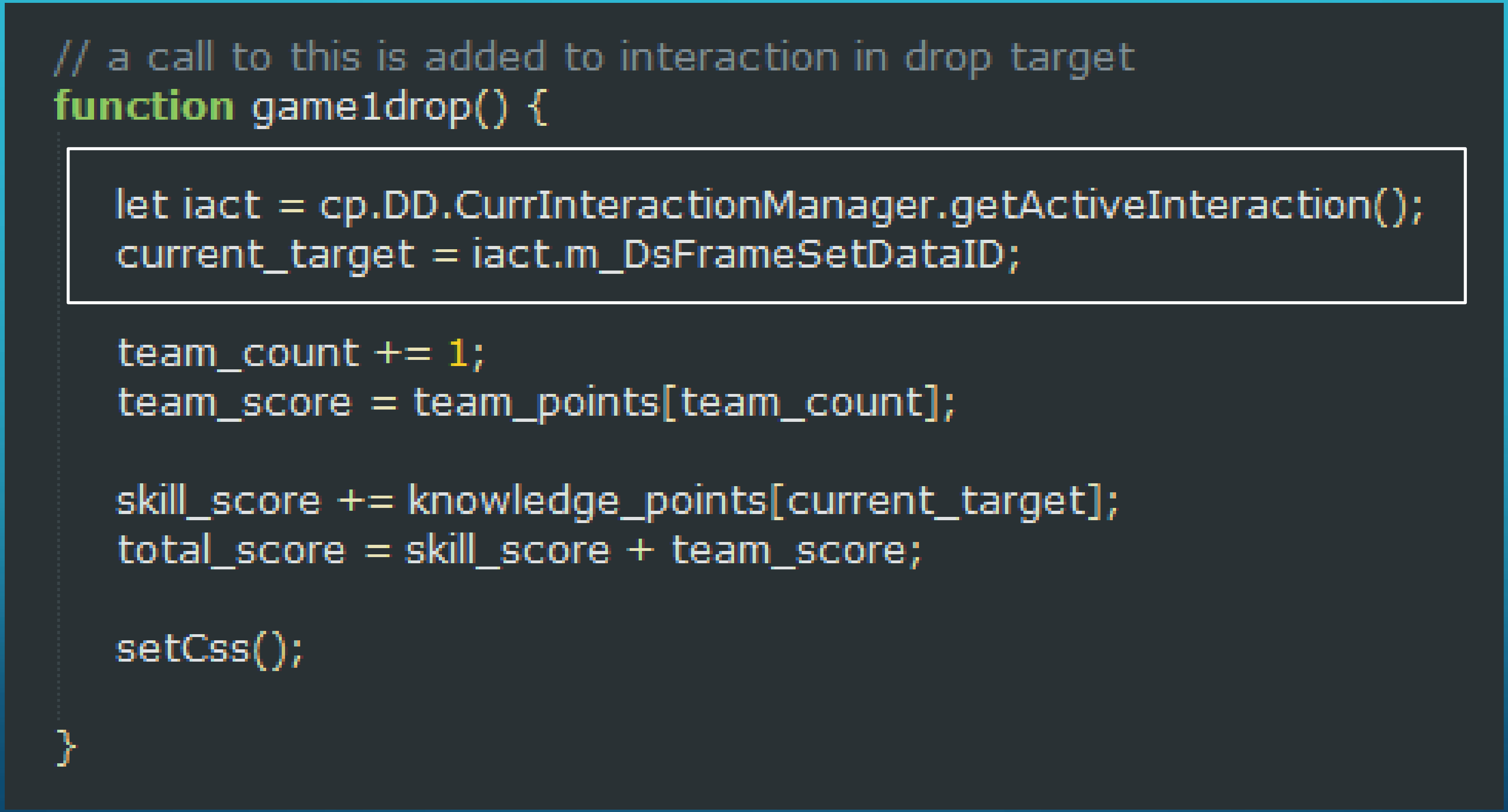
"team" correct
Answer pool



No.	Drop Target	Drag Source	Count
1	team	candidates	16

EXAMPLE – DRAG AND DROP CUSTOM SCORING

HTTPS://GITHUB.COM/SDWARWICK/CAPTIVATE-DEMOS



```
// a call to this is added to interaction in drop target
function game1drop() {

    let iact = cp.DD.CurrInteractionManager.getActiveInteraction();
    current_target = iact.m_DsFrameSetDataID;

    team_count += 1;
    team_score = team_points[team_count];

    skill_score += knowledge_points[current_target];
    total_score = skill_score + team_score;

    setCss();

}
```

Let JavaScript figure out what source item was moved

Create scoring based on some criteria
 - count of dragged components
 - value score for team member

Give feedback by changing colors of shapes directly using CSS on shapes

UNDOCUMENTED CAPTIVATE FUNCTIONS AND DATA STRUCTURES USED IN THESE TWO EXAMPLES

cp.changeState(targetID, state)

cp.getCurrentStateNameForSlideItem(targetID);

cp.show(targetID) , cp.hide(targetID)

cp.D[targetID].qnq (find question data for targetID)

cp.D[questionID].w (question score value – can read and write!)

cp.SubmitInteractions(targetID, cp.QuestionStatusEnum.CORRECT, 0) (click answer button!)

cp.DD.CurrInteractionManager.getActiveInteraction() (get activeDDInteraction)

activeDDInteraction.m_DsFrameSetDataID; (id of last dropped target)

ActiveDDInteraction.OnResetButtonClicked(); (click DD reset button)

activeDDInteraction.undoAvailable (check if undo is available)

activeDDInteraction.OnUndoButtonClicked();

activeDDInteraction.OnSubmitButtonClicked();

cp.RuntimeMessageBox(document.getElementById("cpDocument"), 1) (create a new message box)

UNDOCUMENTED CAPTIVATE FUNCTIONS AND DATA STRUCTURES..

The CPM.js library

- 25,000 JavaScript statements in the basic library to “run” a presentation
- 100,000+ statements to define all objects in a large presentation

CPM.js defines 100+ “top level objects/properties”

CP top object - defines 751 objects/properties

CP.D - all of the slide objects and quizzing information

CP.DD - drag/drop interaction data

CP.em - event manger

CP.movie – timing manager

Lots of other things, too much to even begin to describe..

- Animation
- Display timing
- Quiz handling
- Drag/Drop interactions
- LMS Reporting system..

CPM.js code is well organized with very descriptive top level function names

FAR TOO MUCH TO “FIGURE OUT” IN CPM.JS

Efficient development strategy for web developers

- Build basic shapes and simple interactions using Captivate UI
- HTML/CSS/JAVASCRIPT interactions
- Bridge back into Captivate using the CPM.js library functions
- Use Adobe Documented JavaScript library as starting place
- Leverage undocumented features only as needed

JAVASCRIPT TO CAPTIVATE BRIDGE

- All shape information is found in the object CP.D
  - cp.D.shapename
  - cp.D.shapenamec
  - cp.D.shapenameq0
- Shape name is used as a base to build HTML

  <div id=theSquare>

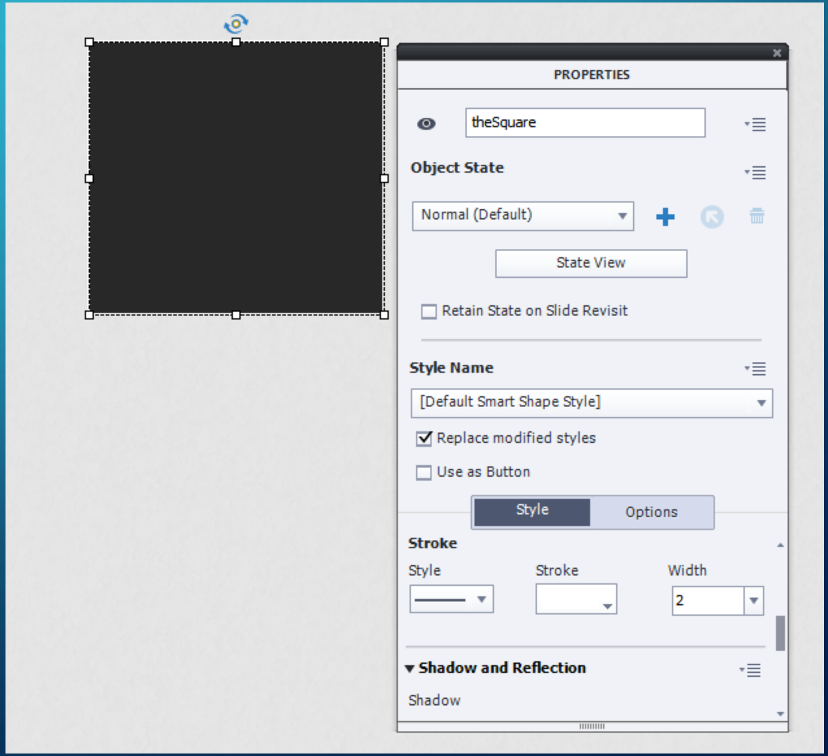
  <canvas id=theSquarec>

  <div id=theSquare_vTxtHolder>

  <div id=re-theSquarec>

  <div id=theSquare_vTxtHandlerHolder>

  <div id=theSquareaccStr>
- Use these objects to create custom effects



PROPERTIES

theSquare

Object State: Normal (Default)

State View

☐ Retain State on Slide Revisit

Style Name: [Default Smart Shape Style]

☑ Replace modified styles

☐ Use as Button

Style | Options

Stroke

Style | Stroke | Width: 2

▾ Shadow and Reflection

Shadow

JAVASCRIPT TO CAPTIVATE BRIDGE

- All variables in captivate are now global JavaScript variables

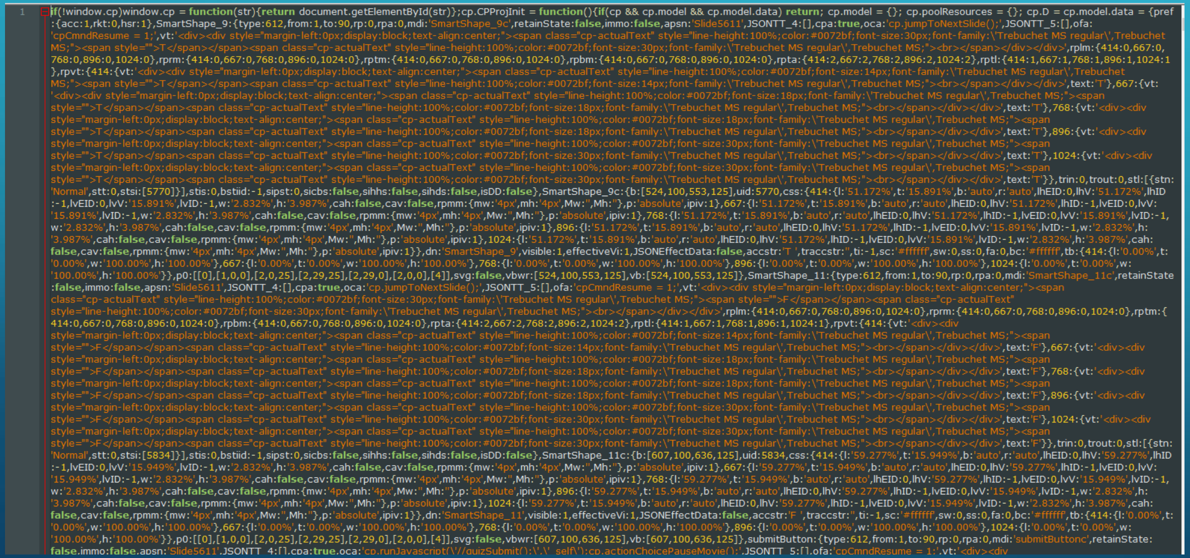
  Example: cpInfoCurrentSlide == cpAPIInterface.getVariableValue("cpInfoCurrentSlide")

- Event-driven JavaScript functions (mouse clicks..)
  - Indirect: use actions and scripts in captivate
  - Direct: use JavaScript events tied directly to HTML objects
- Captivate monitors all variable values once every frame interval (1/30 sec)
  - Simply setting timing-control variables to “true” will cause changes in state
  - Example: cpCmndNextSlide = 1
- Quiz management has another data structure, too much to describe here

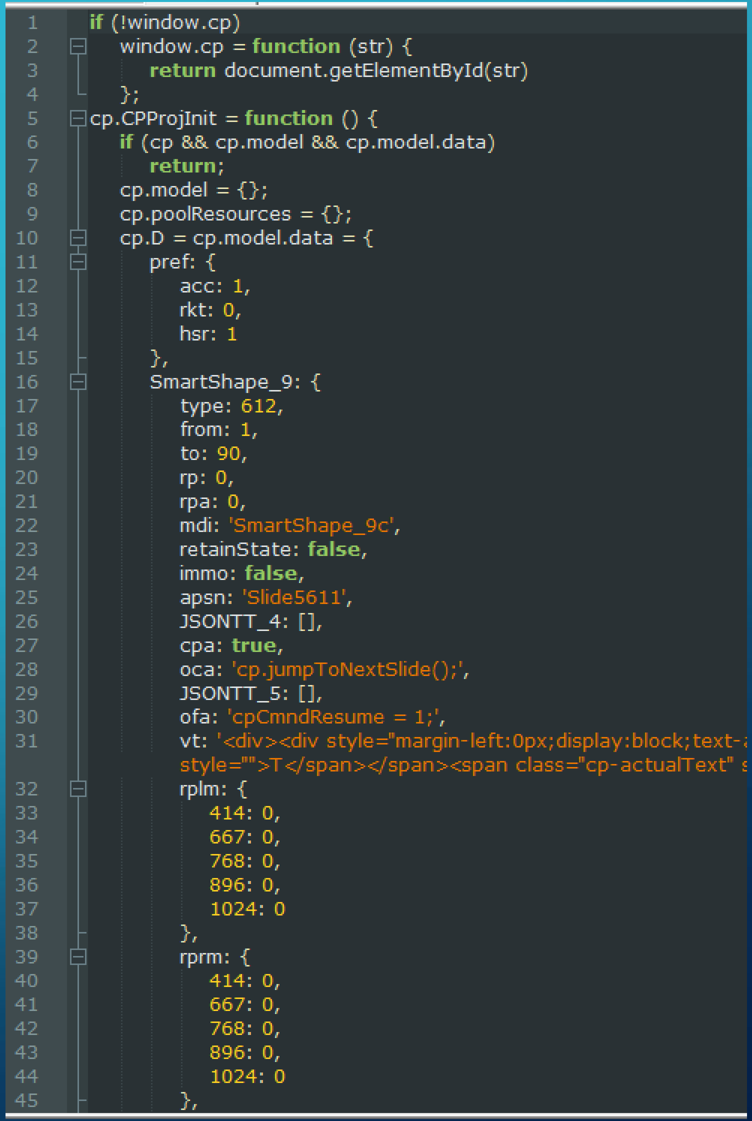
CPM.JS INTERNALS

Notepad++ JavaScript formatter

- Convert compressed CPM.js to readable code
- Save formatted version back into project
- Enables modification & debugging



```
if(!window.cp)window.cp = function(str){return document.getElementById(str)};cp.CPProjInit = function(){if(cp && cp.model && cp.model.data) return; cp.model = {}; cp.poolResources = {}; cp.D = cp.model.data = {pref
:{acc:1,rkt:0,hsr:1},SmartShape_9:{type:612,from:1,to:90,rp:0,rpa:0,mdi:'SmartShape_9c',retainState:false,immo:false,apsn:'Slide5611',JSONTT_4:[],cpa:true,oca:'cp.jumpToNextSlide();',JSONTT_5:[],ofa:
'cpCmndResume = 1;',vt:'<div><div style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet
MS;"><span style="">T</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',rplm:{414:0,667:0,
768:0,896:0,1024:0},rprm:{414:0,667:0,768:0,896:0,1024:0},rptm:{414:0,667:0,768:0,896:0,1024:0},rpbm:{414:0,667:0,768:0,896:0,1024:0},rpta:{414:2,667:2,768:2,896:2,1024:2},rptl:{414:1,667:1,768:1,896:1,1024:1
},rpvt:{414:{vt:'<div><div style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:14px;font-family:\'Trebuchet MS regular\',Trebuchet
MS;"><span style="">T</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:14px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'T'},667:{vt:
'<div><div style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:18px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">T</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:18px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'T'},768:{vt:'<div><div
style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:18px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">T</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:18px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'T'},896:{vt:'<div><div
style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">T</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'T'},1024:{vt:'<div><div
style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">T</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'T'}},trin:0,trout:0,stl:[{stn:
'Normal',stt:0,stsi:[5770]}],stis:0,bstiid:-1,sipst:0,sicbs:false,sihhs:false,sihds:false,isDD:false},SmartShape_9c:{b:[524,100,553,125],uid:5770,css:{414:{l:'51.172%',t:'15.891%',b:'auto',r:'auto',lhEID:0,lhV:'51.172%',lhID
:-1,lvEID:0,lvV:'15.891%',lvID:-1,w:'2.832%',h:'3.987%',cah:false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1},667:{l:'51.172%',t:'15.891%',b:'auto',r:'auto',lhEID:0,lhV:'51.172%',lhID:-1,lvEID:0,lvV:
'15.891%',lvID:-1,w:'2.832%',h:'3.987%',cah:false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1},768:{l:'51.172%',t:'15.891%',b:'auto',r:'auto',lhEID:0,lhV:'51.172%',lhID:-1,lvEID:0,lvV:'15.891%',lvID:-1,
w:'2.832%',h:'3.987%',cah:false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1},896:{l:'51.172%',t:'15.891%',b:'auto',r:'auto',lhEID:0,lhV:'51.172%',lhID:-1,lvEID:0,lvV:'15.891%',lvID:-1,w:'2.832%',h:
'3.987%',cah:false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1},1024:{l:'51.172%',t:'15.891%',b:'auto',r:'auto',lhEID:0,lhV:'51.172%',lhID:-1,lvEID:0,lvV:'15.891%',lvID:-1,w:'2.832%',h:'3.987%',cah:
false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1}},dn:'SmartShape_9',visible:1,effectiveVi:1,JSONEffectData:false,accstr:'T ',traccstr:'',ti:-1,sc:'#ffffff',sw:0,ss:0,fa:0,bc:'#ffffff',tb:{414:{l:'0.00%',t:
'0.00%',w:'100.00%',h:'100.00%'},667:{l:'0.00%',t:'0.00%',w:'100.00%',h:'100.00%'},768:{l:'0.00%',t:'0.00%',w:'100.00%',h:'100.00%'},896:{l:'0.00%',t:'0.00%',w:'100.00%',h:'100.00%'},1024:{l:'0.00%',t:'0.00%',w:
'100.00%',h:'100.00%'}},p0:[[0],[1,0,0],[2,0,25],[2,29,25],[2,29,0],[2,0,0],[4]],svg:false,vbwr:[524,100,553,125],vb:[524,100,553,125]},SmartShape_11:{type:612,from:1,to:90,rp:0,rpa:0,mdi:'SmartShape_11c',retainState
:false,immo:false,apsn:'Slide5611',JSONTT_4:[],cpa:true,oca:'cp.jumpToNextSlide();',JSONTT_5:[],ofa:'cpCmndResume = 1;',vt:'<div><div style="margin-left:0px;display:block;text-align:center;"><span
class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span style="">F</span></span><span class="cp-actualText"
style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',rplm:{414:0,667:0,768:0,896:0,1024:0},rprm:{414:0,667:0,768:0,896:0,1024:0},rptm:{
414:0,667:0,768:0,896:0,1024:0},rpbm:{414:0,667:0,768:0,896:0,1024:0},rpta:{414:2,667:2,768:2,896:2,1024:2},rptl:{414:1,667:1,768:1,896:1,1024:1},rpvt:{414:{vt:'<div><div
style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:14px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">F</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:14px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'F'},667:{vt:'<div><div
style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:18px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">F</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:18px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'F'},768:{vt:'<div><div
style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:18px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">F</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:18px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'F'},896:{vt:'<div><div
style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">F</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'F'},1024:{vt:'<div><div
style="margin-left:0px;display:block;text-align:center;"><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><span
style="">F</span></span><span class="cp-actualText" style="line-height:100%;color:#0072bf;font-size:30px;font-family:\'Trebuchet MS regular\',Trebuchet MS;"><br></span></div></div>',text:'F'}},trin:0,trout:0,stl:[{stn:
'Normal',stt:0,stsi:[5834]}],stis:0,bstiid:-1,sipst:0,sicbs:false,sihhs:false,sihds:false,isDD:false},SmartShape_11c:{b:[607,100,636,125],uid:5834,css:{414:{l:'59.277%',t:'15.949%',b:'auto',r:'auto',lhEID:0,lhV:'59.277%',lhID
:-1,lvEID:0,lvV:'15.949%',lvID:-1,w:'2.832%',h:'3.987%',cah:false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1},667:{l:'59.277%',t:'15.949%',b:'auto',r:'auto',lhEID:0,lhV:'59.277%',lhID:-1,lvEID:0,lvV:
'15.949%',lvID:-1,w:'2.832%',h:'3.987%',cah:false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1},768:{l:'59.277%',t:'15.949%',b:'auto',r:'auto',lhEID:0,lhV:'59.277%',lhID:-1,lvEID:0,lvV:'15.949%',lvID:-1,
w:'2.832%',h:'3.987%',cah:false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1},896:{l:'59.277%',t:'15.949%',b:'auto',r:'auto',lhEID:0,lhV:'59.277%',lhID:-1,lvEID:0,lvV:'15.949%',lvID:-1,w:'2.832%',h:
'3.987%',cah:false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1},1024:{l:'59.277%',t:'15.949%',b:'auto',r:'auto',lhEID:0,lhV:'59.277%',lhID:-1,lvEID:0,lvV:'15.949%',lvID:-1,w:'2.832%',h:'3.987%',cah:
false,cav:false,rpmm:{mw:'4px',mh:'4px',Mw:'',Mh:''},p:'absolute',ipiv:1}},dn:'SmartShape_11',visible:1,effectiveVi:1,JSONEffectData:false,accstr:'F ',traccstr:'',ti:-1,sc:'#ffffff',sw:0,ss:0,fa:0,bc:'#ffffff',tb:{414:{l:'0.00%',t:
'0.00%',w:'100.00%',h:'100.00%'},667:{l:'0.00%',t:'0.00%',w:'100.00%',h:'100.00%'},768:{l:'0.00%',t:'0.00%',w:'100.00%',h:'100.00%'},896:{l:'0.00%',t:'0.00%',w:'100.00%',h:'100.00%'},1024:{l:'0.00%',t:'0.00%',w:
'100.00%',h:'100.00%'}},p0:[[0],[1,0,0],[2,0,25],[2,29,25],[2,29,0],[2,0,0],[4]],svg:false,vbwr:[607,100,636,125],vb:[607,100,636,125]},submitButton:{type:612,from:1,to:90,rp:0,rpa:0,mdi:'submitButtonc',retainState:
false,immo:false,apsn:'Slide5611',JSONTT_4:[],cpa:true,oca:'cp.runJavascript(\'//quizSubmit();\',\'_self\');cp.actionChoicePauseMovie();',JSONTT_5:[],ofa:'cpCmndResume = 1;',vt:'<div><div
```



```
if (!window.cp)
    window.cp = function (str) {
        return document.getElementById(str)
    };
cp.CPProjInit = function () {
    if (cp && cp.model && cp.model.data)
        return;
    cp.model = {};
    cp.poolResources = {};
    cp.D = cp.model.data = {
        pref: {
            acc: 1,
            rkt: 0,
            hsr: 1
        },
        SmartShape_9: {
            type: 612,
            from: 1,
            to: 90,
            rp: 0,
            rpa: 0,
            mdi: 'SmartShape_9c',
            retainState: false,
            immo: false,
            apsn: 'Slide5611',
            JSONTT_4: [],
            cpa: true,
            oca: 'cp.jumpToNextSlide();',
            JSONTT_5: [],
            ofa: 'cpCmndResume = 1;',
            vt: '<div><div style="margin-left:0px;display:block;text-a
            style="">T</span></span><span class="cp-actualText" s
            rplm: {
                414: 0,
                667: 0,
                768: 0,
                896: 0,
                1024: 0
            },
            rprm: {
                414: 0,
                667: 0,
                768: 0,
                896: 0,
                1024: 0
            },
```

WHAT ELSE DOES JAVASCRIPT OPEN UP?

- “Real” Jeopardy-style interactions
- Dynamic content
  - Back-end data sources (AJAX)
- Real-time group interactions
- Video game-level animations

- References to external content
  - Fonts, Script libraries
  - Dynamic Graphing and Charting
- Fine-grained experience measurement
  - Total Quiz / per question timing
- Scoring of embedded web content
  - Pass information between parent/child windows
- Custom reporting to LMS/LRS

Access to the entire web development community!

QUESTIONS?

Steven Warwick, eLearningOcean LLC
sdwarwick@elearningocean.com

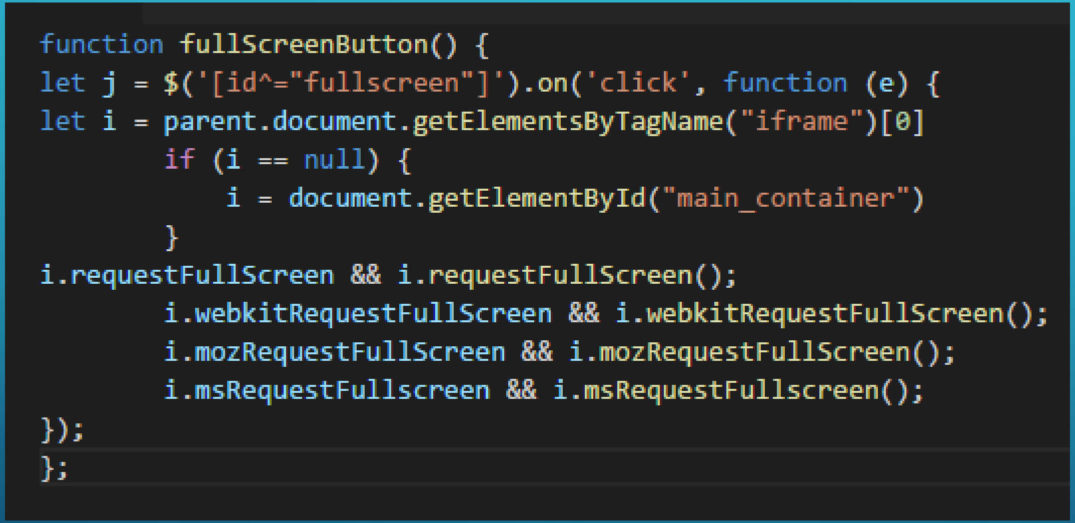
ADDITIONAL MATERIALS

Steven Warwick, eLearningOcean LLC
sdwarwick@elearningocean.com

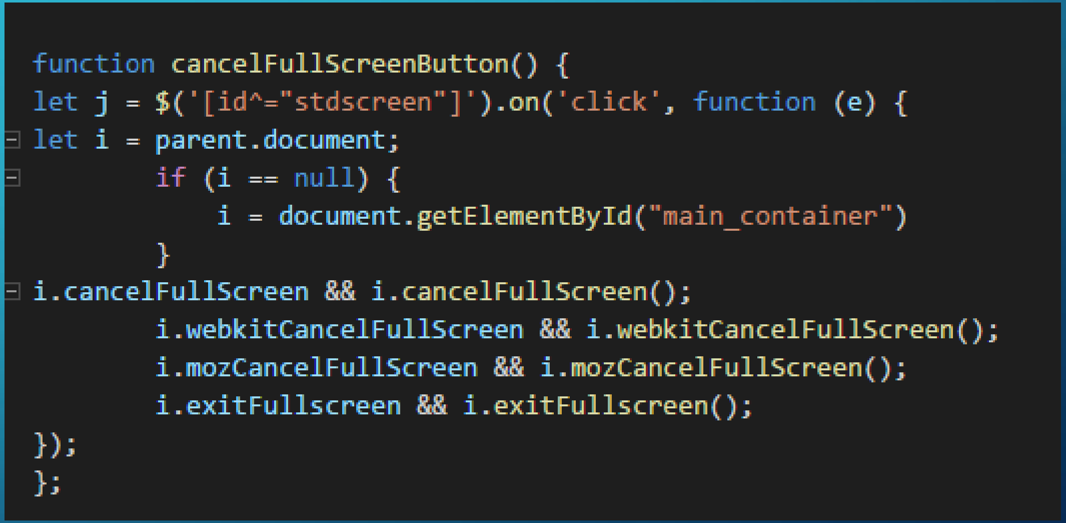
EXAMPLE – “FULL SCREEN” MODE

- Any button that has a name starting in “fullscreen” will activate this code

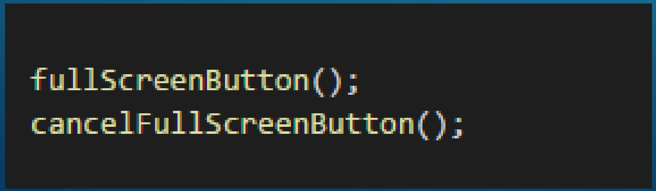
- Also works for presentations embedded in other applications (IFRAME)



```
function fullScreenButton() {
let j = $('[id^="fullscreen"]').on('click', function (e) {
let i = parent.document.getElementsByTagName("iframe")[0]
        if (i == null) {
            i = document.getElementById("main_container")
        }
i.requestFullScreen && i.requestFullScreen();
        i.webkitRequestFullScreen && i.webkitRequestFullScreen();
        i.mozRequestFullScreen && i.mozRequestFullScreen();
        i.msRequestFullscreen && i.msRequestFullscreen();
});
};
```



```
function cancelFullScreenButton() {
let j = $('[id^="stdscreen"]').on('click', function (e) {
let i = parent.document;
        if (i == null) {
            i = document.getElementById("main_container")
        }
i.cancelFullScreen && i.cancelFullScreen();
        i.webkitCancelFullScreen && i.webkitCancelFullScreen();
        i.mozCancelFullScreen && i.mozCancelFullScreen();
        i.exitFullscreen && i.exitFullscreen();
});
};
```



```
fullScreenButton();
cancelFullScreenButton();
```

DOCUMENTED CAPTIVATE/JAVASCRIPT FUNCTIONS

- https://helpx.adobe.com/captivate/using/common-js-interface.html

cpAPIInterface.getVariableValue	Returns the value of the given variable name.
cpAPIInterface.setVariableValue	Sets value of the given variable name
cpAPIInterface.play	Plays the movie.
cpAPIInterface.pause	Pauses the movie.
cpAPIInterface.stop	Stops the movie.
cpAPIInterface.rewind	Rewinds and plays the movie.
cpAPIInterface.next	Seeks the movie to the next slide.
cpAPIInterface.previous	Seeks the movie to the previous slide.
cpAPIInterface.fastForward	Increases the movie speed to 2x, then 4x and then back to normal on consecutive calls.
cpAPIInterface.getPlaySpeed	Returns movie playback speed in Frames per second (fps).
cpAPIInterface.getDurationInFrames	Returns the total number of frames in the movie.
cpAPIInterface.getDurationInSeconds	Returns the total duration of the movie in seconds.
cpAPIInterface.getVolume	Returns the volume of the movie in percentage.
cpAPIInterface.setVolume	Sets the volume of the movie.
cpAPIInterface.navigateToTime	Seeks to a particular time (milliseconds) in the movie.
cpAPIInterface.canNavigateToTime	Returns a boolean value showing whether you can seek to a particular time in the movie or not.
cpAPIInterface.getCurrentFrame	Returns the current frame of the movie.
cpAPIInterface.getCurrentSlideIndex	Returns the current slide index of the movie.
cpAPIInterface.getEventEmitter	Returns the handle to the cpAPIEventEmitter object.

DOCUMENTED CAPTIVATE/JAVASCRIPT EVENTS

- https://helpx.adobe.com/captivate/using/common-js-interface.html

cpAPIEventEmitter.addEventListener (event, function)
cpAPIEventEmitter.removeEventListener(event)
CPAPI_SLIDEENTER
CPAPI_SLIDEEXIT
CPAPI_STARTPLAYBARSCRUBBING
CPAPI_ENDPLAYBARSCRUBBING
CPAPI_INTERACTIVEITEMSUBMIT
CPAPI_MOVIEPAUSE
CPAPI_MOVIERESUME
CPAPI_MOVIESTART
CPAPI_MOVIESTOP
CPAPI_QUESTIONSKIP

105 “TOP LEVEL” VARIABLES GENERATED BY CPM.JS

cp
cpXHRJSLoader
cpAPIInterface
cpAPIEventEmitter
cpCmndVolume
cpCmndMute
cpCmndCC
cpCmndNext
cpCmndNextSlide
cpCmndPrevious
cpCmndNextOnReview
cpCmndPreviousSlide
cpCmndPreviousOnReview
cpCmndPlaybarMoved
cpCmndShowPlaybar
cpCmndFastForward
cpCmndRewindAndPlay
cpCmndRewindAndStop
cpCmndGotoFrame
cpCmndGotoFrameAndResume
cpCmndGotoSlide
cpCmndGotoSlideAndResume
cpCmndGotoSlideByUIDAndResume
cpCmndResume
cpCmndPause
cpCmndExit
cpLockTOC
cpCmndInfo
cpCmndTOCVisible
cpInfoSlidesInProject
cpInfoFPS
cpInfoAuthor
cpInfoCompany
cpInfoEmail
cpInfoWebsite
cpInfoCopyright
cpInfoProjectName
cpInfoDescription
cpInfoCurrentFrame
_cpInfoCurrentFrame
cpInfoPrevFrame
cpInfoFrameCount
cpInfoPrevSlide
_cpInfoPrevSlide
cpInfoLastVisitedSlide
_cpInfoLastVisitedSlide
cpInfoCurrentSlide
cpInfoCurrentSlideIndex
_cpInfoCurrentSlide
cpInfoCurrentSlideLabel
_cpInfoCurrentSlideLabel
cpInfoSlideCount
cpInfoIsStandalone
cpInfoHasPlaybar
cpInfoCurrentSlideType
cpInfoIsResultSlide
cpInfoElapsedTimeMS
cpInfoEpochMS
cpInfoCurrentMinutes
cpInfoCurrentHour
cpInfoCurrentTime
cpInfoCurrentDay
cpInfoCurrentYear
cpInfoCurrentMonth
cpInfoCurrentDate
cpInfoCurrentDateString
cpInfoCurrentDateStringDDMMYYYY
cpInfoCurrentLocaleDateString
cpCmndGotoQuizScopeSlide
cpQuizInfoLastSlidePointScored
cpQuizInfoQuestionSlideType
cpQuizInfoAnswerChoice
cpQuizInfoMaxAttemptsOnCurrentQuestion
cpQuizInfoPointsPerQuestionSlide
cpQuizInfoNegativePointsOnCurrentQuestionSlide
cpQuizInfoQuestionSlideTiming
cpQuizInfoQuizPassPoints
cpQuizInfoQuizPassPercent
cpQuizInfoTotalProjectPoints
cpQuizInfoTotalUnansweredQuestions
cpQuizInfoNoQuestionsPerQuiz
cpQuizInfoPointsscored
cpQuizInfoPretestPointsscored
cpQuizInfoPretestScorePercentage
cpQuizInfoTotalCorrectAnswers
cpInfoPercentage
cpQuizInfoTotalQuizPoints
cpQuizInfoAttempts
cpQuizInfoTotalQuestionsPerProject
cpQuizInfoQuestionPartialScoreOn
cpQuizScopeSlide
cpInQuizScope
cpQuizInfoPassFail
cpInfoCourseID
cpInfoCourseName
cpQuizInfoPreTestTotalCorrectAnswers
cpInReviewMode
cpQuizInfoPreTestTotalQuestions
cpQuizInfoPreTestMaxScore
cpInfoMobileOS
cpQuizInfoStudentID
cpQuizInfoStudentName
cpQuizHandledAll