番号	主題	項目	出題種別	説明（望まれるスキル）	関連キーワード	重要度
2.1	JavaScript					
2.1.1		JavaScript文法	知識問題 コードリーティング問題 記述問題	Javascriptの概要 Javascriptに関する他言語との違いや、一般的な注意事項について理解している。 ・Javascriptの文法および記述方法 ・Javascriptにおけるオブジェクトおよびプロパティの概要 ・ガーベージコレクションの概要 ・strictモードの概要 ・Jvascriptの予約語 ・グローバル関数の概要 ・グローバルオブジェクトの概要	変数、型、オブジェクト型、プリミティブ型 インターフェイス、プロパティ ガーベージコレクション 特殊数値（Infinity,NaN,Number.MAX_VALUE,Number.MIN_VALUE） null, undefined strictモード グローバル関数 (decodeURI(),decodeURIComponent(),encodeURI(),encodeURIComponent(),escape(),eval(),isFinite(),isNaN(),parseFloat(),ParseInt(),unescape()) グローバルオブジェクト (Array,Boolean,Date,Error,JSON、Math,Number,Object,RegExp,String) 算術演算子,等値演算子(==,===,!=,!==),関係演算子(<,>,<=,>=,in,instanceof),論理演算子,ビット演算子,typeof演算子,new,delete演算子,void演算子 switch,break,case,continue,do,while, for,for/in,if/else,throw,try/with オブジェクト プロパティ Objectクラス prototypeプロパティ this スコープ、スコープチェーン、グローバルオブジェクト、Callオブジェクト、with文 クロージャ	10
				演算子 演算子の適切な使い方・他の演算子との使い分けを理解している。 ・算術演算子、比較演算子の種類 ・文字列に対する演算子の使用方法と結果 ・==と===の違い ・instanceof演算子、new演算子、delete演算子、void演算子の意味と結果 ・数値と文字列に対する比較演算子の結果 ・オブジェクト型とプリミティブ型に対する比較演算子の結果 ・正規表現		
				特殊数値 特殊数値の意味を理解しており、他の特殊数値との違いを説明できる事を確認する。 ・特殊数値の意味（Infinity,Nan,Number.MAX_VALUE,Number.MIN_VALUE） ・nullとundefinedの違い		
				配列 多次元配列を含む、配列の生成方法や動作について正しく理解している。 ・配列の生成方法における複数の記述方法 ・配列の長さ変更時の動作（lengthプロパティの書き換え） ・多次元配列の生成方法		
				制御文 Javascriptの制御文について使い方と、利用した場合のプログラムの動きについて理解している。 ・基本制御文（switch,break,case,continue,do,while） ・throw文、try,catch,finallyの動作とエラーオブジェクト ・with文 ・for/inの動作		
				関数 関数の定義方法や匿名関数の利用方法について理解している。 ・関数の定義方法 ・関数の戻り値を指定しなかった場合の動き ・宣言型関数、関数リテラル、関数オブジェクトとしての匿名（無名）関数の動作と定義方法 ・オブジェクトプロパティへの関数挿入		
				型・オブジェクト Jacasvriptにおけるオブジェクトに関する知識・プリミティブ型変数との違いなどを理解している。 ・オブジェクト型の作成と、プリミティブ型の作成方法の違い ・型変換で出力される結果の型がどのようになるのか ・プロトタイプオブジェクトの概要 ・prototypeプロパティを利用した継承方法		
				プロパティ プロパティの追加・削除などの操作方法や、プロトタイププロパティの利用方法について理解している。 ・オブジェクトに対するプロパティの追加方法 ・プロパティへのアクセス方法 ・プロパティの存在確認方法 ・constructorプロパティの意味 ・プロトタイプの定義（クラスメソッドとの違い） ・コンストラクタチェーン（継承） ・deleteによるプロパティ除去とその結果 ・プロトタイプチェーン経由の確認方法（hasOwnProperty()） ・call()、apply()を利用した場合の、thisオブジェクトの使い方		
				スコープ Javascriptにおける宣言場所や呼び出しの記述箇所によるスコープの変化について理解している。 ・変数などの宣言する場所や、記述による変数・関数・イベントハンドラなどのスコープの範囲を理解している。 ・Javascriptにおけるスコープチェーンによる変数定義などの探索のしくみを理解している。 ・with文による、スコープチェーンの変更方法を理解している。 ・クロージャを使ったプライベート変数の定義と、参照・変更を行う関数を定義するコードを記述する事ができる。		
2.2	WebブラウザにおけるJavaScript API					
2.2.1		イベント	知識問題 コードリーティング問題 記述問題	Javascriptのページ読み込みや、ユーザ操作によって発生するイベントの発生タイミングを理解しており、イベント処理を行うコードを記述する事ができる。 ・HTML文書を読み込む際に発生するイベント名とその順序 ・代表的なフォームイベントの概要と使い方 ・代表的なキーボードイベントの概要と使い方 ・代表的なマウスイベントの概要と使い方 ・代表的なタッチイベントの概要と使い方 ・フォームイベントの登録・呼び出しと、入力情報の処理 ・キーボードイベントの登録・呼び出しと、入力情報の処理 ・マウスイベントの登録・呼び出しと、入力情報の処理 ・タッチ系イベントの登録・呼び出しと、入力情報の処理 ・ドラッグアンドドロップイベントの登録・呼び出しと、入力情報の処理 ・スマートフォンにおける回転イベントの登録・呼び出しと、入力情報の処理 ・カスタムイベントの登録・呼び出しと、入力情報の処理 ・イベントリスナの登録、削除	onloadイベント EventTargetインターフェイス メソッド（addEventListener(),dispatchEvent(),removeEventListener()） フォームイベント onblur,onchange,oncontextmenu,onfocus,onformchange,onforminput,oninput,oninvalid,onselect,onsubmit キーボードイベント onkeydown,onkeypress,onkeyup マウスイベント onclick,ondblclick,ondrag,ondragend,ondragenter,ondragleave,ondragover,ondragstart,ondrop,onmousedown,onmousemove,onmouseout,onmouseover,onmouseup,onmousewheel,onscroll タッチ系イベント touchstart,touchmove,touchend bind(),trigger()	8
2.2.2		ドキュメントオブジェクト／DOM	知識問題 コードリーティング問題	イベント発生時などにDOMを利用して、HTMLの内容を読み込む、書き換える処理にを行うコードの記述方法について理解している。 ・要素の親および子要素の取得 ・要素の表示、非表示制御 ・要素の上書き。（innerHTML） ・要素の挿入、削除。 ・属性の追加、取得、削除。 ・フォームのデータにアクセスおよび、入力値の検証。 ・サブミットの中止	DOM document.getElementById document.forms innerHTML createElement() insertBefore() appendChild() createAttribute() hasAttribute() removeAttribute()	6
2.2.3		ウィンドウオブジェクト	知識問題 コードリーティング問題	Windowオブジェクトが持つプロパティについて理解しており、イベント発生時におけるウィンドウの移動などの表示制御を行うコードについて記述方法を理解している ・Windowオブジェクトの概要と仕様 ・プロパティ、メソッド、イベントハンドラの概要 ・プロパティを使った、ウィンドウの座標・大きさなどの確認方法 ・メッセージダイアログの表示 ・ウィンドウやタブのロードおよび指定URLにあるページの表示 ・ウィンドウやタブを閉じる方法 ・スクロールなどのウィンドウ操作 ・postMessageを利用したメッセージ送信とイベント処理 ・setIntervalを利用した繰り返し処理の実行 ・setTimeoutを利用した指定時間後の処理実行 ・イベント処理	Windowオブジェクト ・プロパティ ・コンストラクタ ・メソッド ・イベントハンドラ	8
2.2.4		Selectors API	知識問題 コードリーティング問題	Selectors APIを使ったHTML要素への操作方法についてコードの記述方法を理解している。 ・Selectors APIの概要と提供機能 ・要素の探索 ・取得した要素に対する、追加・変更・削除などの操作	querySelector,querySelectorAll	4
2.2.5		テスト・デバッグ	知識問題	コンソールを使って、変数の内容を出力する方法について理解している。 ・コンソールへのデバッグ出力 ・JavaScriptプロファイラの概要	Consoleオブジェクト メソッド（assert(),debug(),dirxm(),error(),info(),log(),profile(),profileEnd(),trace(),warn()）	2

番号	主題	項目	出題種別	説明（望まれるスキル）	関連キーワード	重要度
2.3	グラフィックス					
2.3.1		Canvas(2D)	知識問題 コードリーティング問題			
				Canvasの特徴について説明ができ、Canvasを使って描画を行うコードを見て、その結果ブラウザ上でどのような動きを行うかを理解する事ができる。 概要 ・Canvasの特徴と提供機能 ・Canvasが利用可能な条件 ・CanvasとSVGの違い コンテキスト ・2Dコンテキストの概要と描画状態の遷移 ・描画状態の保存と復元する方法 ・クリッピング領域を指定し、描画範囲を制限する方法 基本図形描画 ・線、矩形、曲線描画 ・Canvasの塗りつぶし テキスト描画 ・テキスト幅の算定、塗りつぶし描画、輪郭描画 ・フォントの設定 変形（拡大、回転、移動） ・キャンバスの拡大・縮小、回転、移動 エフェクト ・キャンバスへの透明度指定 ・キャンバス上へ図形などを合成 イメージデータ ・イメージの描画処理（伸縮、一部切り取りなどを含む）	canvas.getContext("2d") context.save(),restore() context.beginPath() context.rect(),clip() context.moveTo(),context.lineTo(),context.stroke() context.fillRect(),context.strokeRect(),context.clearRect() context.arc(),context.arcTo(),context.bezierCurveTo(),context.quadraticCurveTo() context.measureText() context.fillText(),context.strokeText() context.font context.setTransform() context.rotate() context.scale() context.translate() context.globalAlpha context.globalCompositeOperation context.drawImage() context.createImageData()	6
2.3.2		SVG	知識問題	SVGの概要とCanvasとの違いについて理解している。 ・SVGの特徴 ・Canvasとの違い	ベクターグラフィック、XML	1
2.4	マルチメディア					
2.4.1		video要素,audio要素 (各要素の仕様や、マークアップの記述方法に関してはレベル1に含まれるため範囲外)	知識問題 コードリーティング問題	動画や音声を再生するコードを呼んで、そのコードの問題点やブラウザ上での動きを理解する事ができる。 ・オーディオデータの再生・停止・状態取得 ・Canvas上での動画表示 ・ビデオデータの再生・停止・状態取得 ・ダウンロード状況に応じた処理 ・メディアリソースの再ロード ・メディアリソースに関するエラーコード取得	HTMLMediaElement(MediaElement)オブジェクト プロパティ（autopkay,controls,currenttime,ended,error,loop,networkState,paused,played,preload,readyState） メソッド（play(),pause(),load()） イベントハンドラ（onplay,onplaying,ontime,onupdate,onpause,onwaiting,onstalled,onended,onerror,onabort）	2
2.5	オフラインアプリケーションAPI					
2.5.1		アプリケーションキャッシュの制御	知識問題 コードリーティング問題	オフラインでも動作可能なアプリケーションを設計するにあたって知っておくべき、状況の確認方法とキャッシュの操作方法について理解をしている。 ・アプリケーションキャッシュの概要 ・アプリケーションキャッシュを利用する場合の注意点 ・ApplicationCachオブジェクトの仕様 ・ブラウザのネット接続状況に関する判別方法	ApplicationCache オブジェクト プロパティ（status） メソッド（update(),swapCache()） イベント（checking,error,noupdate,downloading,progress,updateready,cached,obsolete） Navigatorオブジェクト プロパティ（onLine）	2
2.6	Session History and Navigation					
2.6.1		History API	知識問題 コードリーティング問題	HistoryオブジェクトやLocationオブジェクトが持つプロパティや関数を理解しており、それぞれ要件を実現するためにどれを利用すればいいかを理解できている。 ・Hisotry APIの概要と提供機能 ・History オブジェクトを使った、ページ履歴に関する情報取得 ・History オブジェクトを使った、履歴の操作およびページ移動 ・Historyオブジェクトを使った、ロケーションバー上のURL操作 ・Locationオブジェクトを使った、現在のページに関するURL情報取得 ・Locationオブジェクトを使った、ページのロードおよびリロード ・Locationオブジェクトを使った、ページ履歴の置換	History オブジェクト プロパティ（length） メソッド（go(),back(),forward(),pushState(),replaceState()） Location オブジェクト プロパティ（href,protocol,host,hostname,port,pathname,search,hash） メソッド（assign(),replace(),reload()）	3
2.7	表示制御					
2.7.1		Page Visibility	知識問題	スマートフォンなどでの使用も含めて想定されるHTML5の画面表示制御方法について理解している。 ・Page Visibility機能を使った表示制御の概要 ・ページの表示状態取得 ・表示状態が変化した際のイベント処理	Documentオブジェクト プロパティ（hidden,visibilityState） イベント（visibilitychange）	2
2.7.2		Timing control for script-based a	知識問題 コードリーティング問題	高速な描画処理をブラウザ上で実現するために必要な知識を持ち、処理落ちやちらつきを防ぐための記述方法を理解している。 ・HTML5におけるアニメーションの概要 ・requestAnimationFrameとsetIntervalの違い ・リフレッシュレートとの関係 ・requestAnimationFrameを使ったアニメーションフレーム制御 ・cancelAnimationFrameによるフレーム処理リクエストのキャンセル	Windowオブジェクト メソッド（requestAnimationFrame(),cancelAnimationFrame()） FrameRequestCallbackオブジェクト	2
2.8	ストレージ					
2.8.1		Web Storage	知識問題 コードリーティング問題	Web Storageの特徴や仕様を理解し、読み込み・書き込みを行うコードを理解することができる。 ・Web Storageを利用するアプリケーションを作成するにあたって注意すべき、セキュリティの観点からの注意事項 ・ローカルストレージとセッションストレージの違い ・ローカルストレージに関する仕様と、データの保存、取得、削除 ・セッションストレージに関する仕様と、データの保存、取得、削除 ・ローカルストレージおよびセッションストレージに関するイベント処理	localStorageオブジェクト,sessionStorageオブジェクト Storageオブジェクト プロパティ（length） メソッド（key(),setItem(),getItem(),removeItem(),clear()） StorageEventオブジェクト プロパティ（key,oldValue,newValue,url,storageArea）	4
2.8.2		Indexed Database API	知識問題 コードリーティング問題	Indexed Database APIの特徴を理解し、簡単な読み込み・書き込みを行うコードを理解することができる。 ・Indexed Database APIの特徴と、Web Storageとの違い ・データベースへのアクセス ・データの読み込み ・データの保存	IDBFactoryオブジェクト メソッド（open(),deleteDatabase()） IDBRequestオブジェクト プロパティ（result,error,source,transaction,readyState） IDBDatabaseオブジェクト プロパティ（name,version,objectStoreNames） メソッド（createObjectStore(),deleteObjectStore(),transaction(),close()）	2
2.8.3		File API	知識問題 コードリーティング問題	File APIの特徴を理解し、簡単なファイル読み込みのコードを理解する事ができる。 ・HTML5におけるローカルファイルアクセス機能の概要 ・ローカルファイルアクセスにおけるセキュリティ観点での制限事項 ・ファイルオブジェクトリストの取得 ・ローカルファイルの読み込み	FileListオブジェクト プロパティ（length） メソッド（item()） Blobオブジェクト プロパティ（size,type） メソッド（slice(),close()） Fileオブジェクト プロパティ（name,lastModifiedDate） FileReaderオブジェクト プロパティ（readyState,result,error） メソッド（readAsArrayBuffer(),readAsText(),readAsDataURL(),abort()）	2

番号	主題	項目	出題種別	説明（望まれるスキル）	関連キーワード	重要度
2.9	通信					
2.9.1		WebSocket	知識問題	WebSocketの特徴を理解し、通信を行うにあたって必要な知識について理解している。 ・WebSocketを使った通信の利点と欠点 ・WebSocketにおけるイベント発生タイミング ・WebSocketを使ったサーバとの通信（クライアント側のコード） （WebSocket通信におけるサーバ側のコードについては、試験範囲外とする）	WebSocketオブジェクト メソッド(send(),close()) プロパティ(URL,readyState,bufferedAmount) イベント(onopen,onmessage,onclose)	2
2.9.2		XMLHttpRequest	知識問題 コードリーティング問題	XMLHttpRequestの特徴を理解し、通信を行い結果を適切に処理できるプログラムを読むことができる。 ・XMLHttpRequestを利用した、WebサーバへHTTPリスエストを送信および結果の受信 ・XMLHttpRequestオブジェクトと通信時に関連するイベントハンドラ ・XMLHttpRequestにおけるリクエストヘッダの設定 ・XMLHttpRequestオブジェクトのステータス確認 ・レスポンスデータに関する内容の確認、および用途にあった処理 ・取得データのブラウザによるキャッシュを防ぐ対策	XMLHttpRequest　オブジェクト リクエスト/レスポンス共通 プロパティ(readyState) イベント（onreadystatechange） リクエスト関連 メソッド(open(),setRequestHeader(),send(),abort()) プロパティ(timeout, withCredentials, upload) レスポンス処理 メソッド(getResponseHeader(), getAllResponseHeaders(), overrideMimeType()) プロパティ(status, statusText, responseType, response, responseText, responseXML) XMLHttpRequestEventTarget　Interface イベント（onloadstart,onprogress,onabort,onerror,onload,ontimeout,onloadend）	4
2.10	Geolocation API					
2.10.1		Geolocation APIの基本と位置情報	知識問題	Geolocation APIの概要と利用時の注意点について理解している。 ・Geolocation APIの特徴と注意する点 ・端末における現在の位置情報を取得する方法 ・現在位置取得後のコールバック関数呼び出し	GPS Position オブジェクト プロパティ（cords,timestamp） Coordinates オブジェクト プロパティ(latitude,longitude,accuracy,altitude,altitudeAccuracy,heading,speed) Geolocationオブジェクト メソッド（getCurrentPosition(),watchPosition(),clearWatch()） コールバック(PositionCallback,PositionErrorCallback) PositionErrorオブジェクト プロパティ（code,message）	2
2.11	Web Workers					
2.11.1		並列処理の記述	知識問題 コードリーティング問題	Web Workersの特徴を理解し、並列処理やエラーの検出を行うコードを読みどのように動作するか理解する事ができる。 ・Web Workersの特徴と利用するメリット ・Workerの新規作成 ・メッセージの送受信 ・Worker内での処理に関する注意事項 ・受け渡しパラメータに対するメモリ使用量 ・エラー検知、およびエラーイベント	Workerオブジェクト メソッド(terminate(),postMessage()) イベント（onerror,onmessage） ErrorEventインターフェイスのプロパティ(message,filename,lineno)	4
2.12	パフォーマンス					
2.12.1		Navigation Timing	知識問題	Navigation Timing APIを使って、発生している性能に関する問題を解決するための知識について理解できている。 ・ユーザアクションに対する発生時刻の取得 ・画像の読み込み時間の計測 ・ページの読み込み時間、DNSにおける名前解決などの各所要時間の計測	PerformanceTimingオブジェクト プロパティ(navigationStart, unloadEventStart, unloadEventEnd, redirectStart, redirectEnd, fetchStart, domainLookupStart, domainLookupEnd, connectStart, connectEnd, secureConnectionStart, requestStart, responseStart, responseEnd, domLoading, domInteractive, domContentLoadedEventStart, domContentLoadedEventEnd, domComplete, loadEventStart, loadEventEnd) Performanceオブジェクト プロパティ（timing, navigation） Windowオブジェクト プロパティ（performance）	4
2.12.2		High Resolution Time	知識問題	High Resolution Time APIを使って、高い精度の時間を取得するために必要な知識について理解している。 ・High Resolution Time APIの特徴と提供機能 ・High Resolution Time APIを利用した、高い精度のパフォーマンス測定	Performanceオブジェクト メソッド（now()）	1