Webクリエイター能力認定試験 【HTML Living Standard対応】 出題範囲

2026年10月1日

本試験はHTML Living Standardに準拠し、技術動向に合わせて定期的に改定されます。

※弊社策定のHTML LivingStandard、CSSの機能で出題する。（2026年10月1日更新時点）

※試験問題で編集対象外となるJavaScriptファイル、ノーマライズCSSファイルの内容は出題範囲に含まれません。

※エキスパートはスタンダードの出題範囲を含む。

項目	内容	（備考）関連する拡張子、要素、セレクタ、プロパティなど	スタンダード	エキスパート
■Web制作の基礎知識				
ファイルと役割	基本的なファイルの操作ができる。	-	●	●
ファイルと役割	Webサイトで使用するファイルの種類と役割を理解している。	各種画像・メディアファイル形式 gif、jpeg、png、svg、webp、webmなど	●	●
ソース編集／ファイルの操作	Web制作のための作業環境の構築と、使うツールの基本的な操作ができる。	-	●	●
Webサイトを閲覧できる環境	Webサイトを閲覧するアプリの基本的な機能と、種類を理解している。	-	●	●
Webサイトを閲覧できる環境	ブラウザの開発ツールがどういうものか理解し、HTML/CSSソースコードを確認できる。	-	●	●
文字コード	文字コードがどういうものか理解している。	meta charsetの指定、関連する値	●	●
URLとパス	URLの役割や書かれている内容を理解し、適切なパスを記述できる	相対パスによるURL、画像ファイルなどの指定	●	●
■Webデザイン				
Webサイトの設計と構造	Webサイトの典型的なページ構成を理解している。	-	●	●
Webサイトの設計と構造	HTMLサイトマップの役割と意味を理解している。	-	●	●
Webページのレイアウトと構成要素	Webページの典型的なレイアウトと構成要素、各部名称を理解している。	-	●	●
ビジュアルデザイン	Webサイト上で色を表現する仕組みを理解している。	名前(blue、grayなど）、カラーコード(#)、rgb()、hsl()	●	●
ビジュアルデザイン	配色の基礎的な理論を理解している。	-		●
アクセシビリティ	アクセシビリティがどういうもので、考慮する必要があるのかを理解している。	lang属性、alt属性（代替テキスト）の指定 <label>、<button>などの適切な使用		●
■HTMLの基本構造と概念				
基本構造	HTMLの役割と仕組みを理解している。	-	●	●
基本構造	HTML要素の記述方法を理解し、正しく記述できる。	-	●	●
基本構造	HTMLドキュメントは、要素によって階層構造を作ることを理解している。	-	●	●
基本構造	要素によって子要素にできる要素が違うことを理解している。	-	●	●
基本構造	HTMLコメントを正しく記述できる。	<!-- -->	●	●
■HTML要素				
ドキュメントとメタデータ	HTMLドキュメントを構成する基本要素を理解している。	<html>、<head>、<body>、<title>、<meta>、<script>、関連する属性・値	●	●
ドキュメントとメタデータ	HTMLドキュメントのメタデータを正しく記述できる。	<meta charset>、<link>、<title>、<style>、関連する属性・値	●	●
要素の意味	HTMLに定義されている各要素の意味を理解し、適切に使用できる。	・見出し関連　<h1>～<h6> ・リスト関連　 、 、 、<dl>、<dt>、<dd>など ・ページ構造・セクション関連　<header>、<nav>、<main>、<section>、<article>、<aside>、<footer>など ・テーブル／表関連　<table>、<caption>、<thead>、<tbody>、<tfoot>、<tr>、<th>、<td>など、関連する属性・値 （エキスパートのみ）<fieldset>、<legend> ・フォーム関連　<form>、<label>、<input>、<select>、<option>、<textarea>、<button>など、関連する属性・値 ・テキスト関連　、、<small>、、<time>、<u>など ・その他の主要な要素　<p>、 、 など	●	●
リンクとメディア	適切な要素を使用して、画像・音声・動画メディアおよび別のHTMLファイルを表示することができる。	<img>、<audio>、<video>、<source>、<object>、<figure><picture>、<map>、<area>など、関連する要素・属性・値	●	●
リンクとメディア	HTMLドキュメント中にハイパーリンクを作成し、関連属性を使用して適切な動作を設定できる。	<a>、href属性、target属性	●	●
グループ化・構造化	複数の要素を用いて階層構造を作ることができる。	-	●	●
■CSSの基礎知識				
基本構文と動作原理	WebデザインにおけるCSSの役割と仕組みを理解している。	-	●	●
基本構文と動作原理	CSSルールセットの基本構文と各部の名称・役割を理解し、記述できる。	-	●	●
基本構文と動作原理	カスケード規則と詳細度を理解している。	!important		●
基本構文と動作原理	ネスト記法を理解し、正しく記述できる。	-		●

■CSSコーディング				
セレクタ	適切なセレクタを選択して要素を選択し、必要な場所にスタイルを適用できる。	・ユニバーサルセレクタ、基本セレクタ（クラス、idセレクタ）、子孫セレクタ、子セレクタ、隣接セレクタ、弟セレクタ、属性セレクタ ・疑似クラス　:hover、:active、:focus、:visited、:link、:nth-child、:nth-last-child、:nth-first-child、:nth-of-type、:checked、:required　など （エキスパートのみ）:last-child、:first-child ・疑似要素　::before、::after （エキスパートのみ）::first-line、::first-letter ・関数　（エキスパートのみ）:not()	●	●
プロパティ	デザインカンプを見て、適切なプロパティを選んでスタイルを設定できる。	-	●	●
値と単位	デザインカンプおよびプロパティに応じて正しい値を設定できる。	-	●	●
値と単位	値に指定する単位を理解し、適切に使い分けることができる。	・px、em、rem、%、s、pt ・（エキスパートのみ）vw、vhなど	●	●
@ルール	@ルールの役割を理解している。	@media、@charset、@import	●	●
@ルール	必要に応じて@ルールを記述できる。	@media (max-width)、(min-width)、関連した属性・値	●	●
テキストの装飾	フォントサイズ、ウェイト、スタイル、テキスト色を設定し、テキストの表示を変えられる。	font-size、font-weight、font-style、color、font-familyなど、関連する値	●	●
テキストの装飾	その他テキストの装飾ができる（装飾線を引く・シャドウを設定する・リストマークを削除する）。	text-decoration、list-style、list-style-type、list-style-image、list-style-positionなど、関連する値	●	●
ボックスモデルと背景	ボックスモデルを理解している。	margin、padding、border、width、height、box-sizingなど、関連する値	●	●
ボックスモデルと背景	レスポンシブデザインに適したボックスモデルにするために設定するプロパティと値を理解し、記述できる。	border-box、max-width、min-width、max-height、min-heightなど、関連する値	●	●
ボックスモデルと背景	ボックスの幅と高さを設定できる。	width、height、min-width、max-width、min-height、max-height、aspect-ratioなど、関連する値	●	●
ボックスモデルと背景	ボックスの背景色を設定できる。	background-color、background、関連する値	●	●
ボックスモデルと背景	ボックスに1つまたは複数の背景画像を表示させ、繰り返しや配置の設定ができる。	background-image、background-repeat、background-positionなど、関連する値	●	●
ボックスモデルと背景	ボックスの大きさとは異なるサイズの背景画像を表示でき、かつ伸縮の設定ができる。	background-size、backgroundなど、関連する値	●	●
ボックスモデルと背景	ボックスにグラデーションを適用できる。	linear-gradient()など、関連する値		●
ボックスモデルと背景	ボックスの角を丸くするプロパティを使用できる。	border-radiusなど、関連する値	●	●
ボックスモデルと背景	ボックスにシャドウをかけるプロパティを使用できる。	box-shadow、関連する値	●	●
ボックスモデルと背景	ボックスからコンテンツがあふれたときの表示を制御できる。	overflow、overflow-x、overflow-y、関連する値	●	●
画像とメディアの装飾	縦横比を維持しながら、異なるサイズの親要素に画像をはめ込むことができる。	object-fit、object-position、aspect-ratio、width、height、関連する値	●	●
ページ全体の設定	リセットCSS／ノーマライズCSSの役割を理解している。	-	●	●
ページ全体の設定	ページ全体の背景色・背景画像を設定できる。	background-color、background-image、background	●	●
ページ全体の設定	ページのマージンをゼロに設定できる。	margin、paddingの上下左右の指定含む設定、関連する値	●	●
レイアウト	要素を重ねて配置できる。	positionプロパティ（absolute、fixed、relative、staticなど）、z-index、関連する値	●	●
レイアウト	重ねた要素の位置を調整できる。	・位置指定関連プロパティtop、bottom、left、right、z-index	●	●
レイアウト	フレックスボックスの機能を使用して、要素の配置を調整できる。	・displayプロパティの指定（flex） ・方向・折り返し　flex-direction、flex-wrap ・アイテムのサイズ、伸縮 flex、flex-basis、flex-grow、flex-shrink ・順序　order ・アイテム・コンテナの整列　justify-content、align-items、align-content、align-self ・行と列の間隔　gap、row-gap、column-gap	●	●
レイアウト	グリッドレイアウトの機能を使用して、要素の配置を調整できる。	・displayプロパティの指定（grid） ・グリッドの定義　grid-template-columns、grid-template-rows、grid-template-areas ・グリッドアイテムの配置　grid-column、grid-row、grid-area ・アイテム・コンテンツの整列：justify-items、align-items、justify-self、align-self、justify-content、align-content ・行と列の間隔　gap、row-gap、column-gap		●
テーブルに特有のCSS	テーブルに罫線を引いたり、セル内のコンテンツの行揃えを設定できる。	border、border-collapse、border-spacing、text-align、line-height、text-indentなど、関連する値	●	●
フォームに特有のCSS	テキストフィールド、チェックボックス、ラジオボタン、ボタンのスタイルを変更できる。	・フォームに関連する疑似クラス :checked、:enabled、:disabled、:read-only、:required ・フォームに関連するCSSプロパティ appearance、accent-color	●	●
フォームに特有のCSS	ラベルと各種入力部品の整列ができる。	-	●	●

ボックスの配置と変形	ボックスをページに対して固定した位置に表示できる。	position: fixed、関連する位置指定の値	●	●
ボックスの配置と変形	ページをスクロールしたときに、ボックスを特定の位置で固定することができる。	position: sticky、関連する位置指定の値	●	●
ボックスの配置と変形	ボックスのコンテンツを2次元移動・拡大縮小・回転・歪み変形ができる。	・移動 translate()、translateX()、translateY() ・回転 rotate()、rotateX()、rotateY() ・拡大縮小 scale()、scaleX()、scaleY() ・歪み・傾斜（エキスパートのみ） skew()、skewX()、skewY()	●	●
レスポンシブデザイン	メディアクエリを使用してブレイクポイントを設定できる。	@media (max-width)、@media (min-width)	●	●
レスポンシブデザイン	製作技法としてPCファースト／モバイルファーストを理解している。	-		●
レスポンシブデザイン	画面サイズに合わせて伸縮する要素を作成できる。	widthの設定、max-width、min-widthなど、関連する単位	●	●
レスポンシブデザイン	画面サイズに合わせて伸縮する画像を作成できる。	・widthとheightの設定、max-widthなど ・（エキスパートのみ）aspect-ratio、object-fit、object-position、clamp()	●	●
レスポンシブデザイン	フレックスボックスを使用して、画面サイズに合わせてレイアウトを切り替えられる。	-	●	●
レスポンシブデザイン	グリッドレイアウトを使用して、画面サイズに合わせてレイアウトを切り替えられ	-		●
アニメーション	トランジションを使用して、要素にアニメーションを設定できる。	transition、transition-delay、transition-behaviorなど、関連する値		●
アニメーション	キーフレームアニメーションを設定できる。	・アニメーション設定 @keyframes、animationプロパティ ・関連するプロパティ animation-duration、animation-timing-function、animation-delay、animation-iteration-count、animation-fill-modeなど		●
アニメーション	スプライトシートを使ったアニメーションを設定できる。	・背景画像の指定、調整 background-image、background-position、background-size		●