

Web 全栈开发 职业技能等级标准

(2021 年 1.0 版)

腾讯科技(深圳)有限公司 制定

2021 年 3 月 发布

目 次

前言.....	1
1 范围.....	2
2 规范性引用文件.....	2
3 术语和定义.....	2
4 适用院校专业.....	4
5 面向职业岗位（群）.....	5
6 职业技能要求.....	5
参考文献.....	12

前 言

本标准按照 GB/T 1.1-2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》的规定起草。

本标准起草单位：腾讯科技（深圳）有限公司。

本标准主要起草人：王辉、江源、魏蒙、郭亮、胡菡、陈金朋、赵秉强、张梓雄、张鹏。

声明：本标准的知识产权归属于腾讯科技（深圳）有限公司，未经腾讯科技（深圳）有限公司同意，不得印刷、销售。

1 范围

本标准规定了 Web 全栈开发职业技能等级对应的工作领域、工作任务及职业技能要求。

本标准适用于 Web 全栈开发职业技能培训、考核与评价，相关用人单位的人员聘用、培训与考核可参照使用。

2 规范性引用文件

下列文件对于本标准的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本标准。凡是不注日期的引用文件，其最新版本适用于本标准。

GB-T 5271.1-2000 信息技术 词汇 第1部分：基本术语

GB/T 18792-2002 信息技术 文件描述和处理语言 超文本置标语言 (HTML)

GB_T 37668-2019 信息技术 互联网内容无障碍可访问性技术要求与测试方法

3 术语和定义

GB-T 5271.1-2000、GB_T 37668-2019、GB/T 18792-2002 界定的以及下列术语和定义适用于本文件。

3.1 网页 web page

包含超文本标记语言标签的纯文本文件。

注：网页是构成网站的基本要素。

[GB/T 37668-2019, 定义 2.1]

3.2 超文本标记语言第五版 hyper text markup language 5

通过标记符号来标记要显示的网页中各个部分的一种规范。

注：2014年超文本标记语言完成了第五次重大修改(亦被称为HTML5),用以取代1999年制定的HTML 4.01、XHTML 1.0标准。

[GB/T 37668-2019, 定义 2.9]

3.3 用户界面 user interface

允许信息在个人用户与计算机系统的硬件或软件部件间传送的接口。

注:用户界面,是指对软件的人机交互操作逻辑、界面美观的整体设计。

3.4 浏览器 browser

一个用户代理,其主要功能是把文件呈现给用户。

[GB/T 18792-2002, 定义 4.1]

3.5 超链接 Hyperlink

称为源锚点和宿锚点的两那个锚点间的一种链接关系。该链接从源锚点走向宿锚点。

源锚点也叫做链尾,宿锚点也叫做目的地和链头。

[GB/T 18792-2002, 定义 4.12]

3.6 网络 network

结点和互连分支的一种安排。

[GB-T 5271.1-2000, 定义 01.01.44]

3.7 数据库 database

按照概念结构组织的数据的汇集,它描述这些数据的特征及与数据对应的实体间的关系,并支持一个或多个应用领域。

[GB-T 5271.1-2000, 定义 01.08.05]

3.8 信息安全 information security

保护、维持信息的保密性、完整性和可用性,也可包括真是性、可核查性、抗抵赖性、可靠性等性质。

3.9 布局 layout

应用程序内容中所有对组件进行规划和安排的有关事项。

[GB/T 37668-2019, 定义 2.18]

3.10 组件 component

软件系统中具有相对独立功能、可独立部署、可组装的软件实体。

[GB/T 37668-2019, 定义 2.17]

下列缩略语适用于本文件。

HTML:超文本置标语言(Hyper Text Markup Language)

HTML5:超文本置标语言第五版(Hyper Text Markup Language 5)

URL:统一资源定位符,也称网页地址(Universal Resource Locator)

HTTP:超文本传输协议(HyperText Transfer Protocol)

HTTPS:安全套接字层的超文本传输协议(HyperText Transfer Protocol over Secure Socket Layer)

XSS:跨站脚本(Cross Site Scripting)

CSRF:跨站请求伪造(Cross Site Request Forgery)

API: 应用程序接口 (Application Programming Interface)

4 适用院校专业

中等职业学校: 计算机应用、计算机网络技术、网站建设与管理、软件与信息服务、电子与信息技术、电子技术应用等专业。

高等职业学校: 计算机应用技术、计算机网络技术、软件技术、软件与信息服务、计算机信息管理、计算机系统与维护、信息安全与管理、云计算技术与应用、电子信息工程技术、智能终端技术与应用、移动互联应用技术等专业。

应用型本科学校：计算机科学与技术、软件工程、网络工程、物联网工程、电子科学与技术、电子信息工程等专业。

5 面向职业岗位（群）

主要面向信息技术领域、软件开发公司、互联网公司、大数据公司、软件运维公司等 IT 类公司、企事业单位、政府部门等的信息管理与服务部门，从事 PC 端网页、移动端网页、小程序等可视化、页面和交互的开发工作，从事 PC 端网页、移动端网页、小程序等可视化、页面和交互的开发工作，从事全栈开发、网页设计、数据库开发、API 开发、Web 安全问题解决，代码和性能优化等岗位。

6 职业技能要求

6.1 职业技能等级划分

Web 全栈开发职业技能等级分为三个等级：初级、中级、高级，三个级别依次递进，高级别涵盖低级别职业技能要求。

【Web 全栈开发】（初级）：主要面向 IT 互联网行业，从事各企事业单位、政府信息化数字化部门的 PC 端网页、移动端网页、小程序等前端开发岗位，主要完成各类前端开发、可重用的界面组件编写，还原展示界面等工作，高效完成产品动态信息展现。

【Web 全栈开发】（中级）：主要面向 IT 互联网行业，从事各企事业单位、政府信息化数字化部门的 PC 端网页、移动端网页、小程序等前端开发岗位，主要完成技术选型、架构设计、PC 端和移动端、小程序等可视化开发、产品核心模块、核心代码的编写、页面功能和交互开发 Bug 修复等工作，不断提升用户体验。

【Web 全栈开发】（高级）：主要面向 IT 互联网行业，从事各企事业单位、政府信息化数字化部门 PC 端网页、移动端网页、小程序、全栈开发、架构设计等前端开发岗

位，主要完成相关 API 开发、关系型数据库和 NoSQL 数据库相关开发、Web 安全问题解决、全栈开发、移动端开发、架构设计、性能优化等工作，持续优化代码和性能。

6.2 职业技能等级要求描述

表 1 Web 全栈开发职业技能等级要求（初级）

工作领域	工作任务	职业技能要求
1. 网页开发基础	1.1 网页结构搭建	<p>1.1.1 能够理解 HTML 对于网页的意义。</p> <p>1.1.2 能够了解 HTML 常用标签和属性，掌握超链接、列表、表格、嵌套规则等，能完成页面结构搭建。</p> <p>1.1.3 能够理解 HTML5 与 HTML 区别。</p> <p>1.1.4 能够理解 HTML5 语义化标签，能够结合 HTML 标签搭建网页，提高网页可读性。</p>
	1.2 网页样式和布局	<p>1.2.1 能够理解 CSS 对于网页的意义。</p> <p>1.2.2 能够掌握 CSS 常用语法，选择器的使用等，快速实现样式设置。</p> <p>1.2.3 能够深入理解盒子模型，常见布局方法，灵活运用实现网页布局。</p> <p>1.2.4 能够结合 HTML 标签与 CSS 属性，独立把效果图还原成静态代码页面。</p>
	1.3 网页特效和交互	<p>1.3.1 能够理解 CSS 与 CSS3 区别。</p> <p>1.3.2 能够掌握 CSS3 新增选择器和新增样式属性，能绘制圆角等特殊图形，能掌握元素扭曲、移位、旋转等。</p> <p>1.3.3 能够理解 JavaScript 对于网页的意义。</p> <p>1.3.4 能够掌握 JavaScript 基本语法、DOM 操作和 BOM。</p> <p>1.3.5 能够了解雪碧图与标注设计图，能了解 PS 切图。</p> <p>1.3.6 能够结合 HTML、CSS3 动画和过渡、JavaScript，为网页添加交互行为，高效编写网页特效。</p>
2. 组件化网页开发	2.1 模块和库	<p>2.1.1 能够掌握 JS 函数的高级应用使用面向对象思想进行 DOM 编程，进行模块开发。</p> <p>2.1.2 能够理解 JS 库的特点，掌握主流 JS 库 jQuery 中选择、插件、事件和动画的功能。</p> <p>2.1.3 能够掌握 jQuery 中扩展方法及部分常见插件的使用。</p> <p>2.1.4 能够掌握 jQuery 中自定义插件的方法。</p>
	2.2 浏览器和网络协议	<p>2.2.1 能够了解浏览器的事件机制和渲染过程。</p> <p>2.2.2 能够理解网络体系结构，熟悉 HTTP 协议相关概念。</p> <p>2.2.3 能够理解原生 AJAX 请求流程与细节，掌握常见跨域技巧。</p> <p>2.2.4 能够基于 jQuery 的 AJAX 相关 API，实现异步刷新，异步获取数据的使用方法。</p>

工作领域	工作任务	职业技能要求
	2.3 组件化基础	<p>2.3.1 能够理解变量与作用域，掌握作用域的规则，掌握 JS 函数的高级应用。</p> <p>2.3.2 能够熟练使用面向对象思想进行 DOM 编程，封装相关代码。</p> <p>2.3.3 能够理解组件化的概念、应用场景和基本思路。</p> <p>2.3.4 能够结合页面布局、用户界面、数据、功能等，掌握抽象和封装，实现组件化。</p>
3. 移动端网页开发	3.1 移动端基础	<p>3.1.1 能够了解移动端的概念以及与 PC 端的区别。</p> <p>3.1.2 能够了解移动端屏幕、移动端浏览器、操作系统的不同。</p> <p>3.1.3 能够了解主流移动端页面开发技术选型与常用适配方案。</p> <p>3.1.4 能够了解和解决移动端固定定位、字体单位选择、光标颜色等常见问题。</p>
	3.2 响应式	<p>3.2.1 能够掌握 HTML5 多媒体技术在移动端页面上的应用。</p> <p>3.2.2 能够掌握响应式布局开发与 Flex 布局，掌握 CSS 响应式方案。</p> <p>3.2.3 能够通过网页布局、样式和业务逻辑处理，实现适配不同终端的页面。</p> <p>3.2.4 能够掌握媒体查询中的媒体类型以及媒体特性的概念与用法，能够使用媒体查询实现响应式布局效果。</p>
	3.3 小程序开发	<p>3.3.1 能够了解小程序的概念、开发环境搭建和配置。</p> <p>3.3.2 能够掌握小程序基础组件与操作。</p> <p>3.3.3 能够熟悉小程序基础 API。</p> <p>3.3.4 能够综合应用网页开发技巧和小程序基础知识，独立开发小程序。</p>

表 2 Web 全栈开发职业技能等级要求（中级）

工作领域	工作任务	职业技能要求
1. 性能和工程化	1.1 版本控制	<p>1.1.1 能够使用 Git 进行简单的版本管理。</p> <p>1.1.2 能够使用 Git 进行 pull/push 等版本控制基本操作。</p> <p>1.1.3 能够使用 Git 进行 commit 等提交操作。</p> <p>1.1.4 能够使用 Git 进行分支管理。</p>
	1.2 模块化和包管理器	<p>1.2.1 能够理解作用域、命名空间、模块化之间的联系，能够了解模块化的优点。</p> <p>1.2.2 能够了解模块化方案的进化史和主流的模块化规范。</p> <p>1.2.3 能够了解 Node.js 的作用和特点，掌握 Node.js 的安装，能运行简单的 Node.js 程序。</p> <p>1.2.4 能够理解包管理器，掌握 NPM 源切换，掌握常用命令。</p>

工作领域	工作任务	职业技能要求
	1.3 网页性能	<p>1.3.1 能够理解网页性能的重要性。</p> <p>1.3.2 能够了解网页性能衡量标准和主要的指标。</p> <p>1.3.3 能够使用主流工具对相应网页指标进行分析，并发现影响网页性能的问题点。</p> <p>1.3.4 能够了解网页性能优化的常用手段。</p>
	1.4 打包机制和工程化	<p>1.4.1 能够了解 Webpack 的产生背景，理解其解决的核心问题。</p> <p>1.4.2 能够掌握 Webpack 的安装和基本使用。</p> <p>1.4.3 能够掌握 Webpack 的核心概念和常用配置。</p> <p>1.4.4 能够理解工程化的概念和必然性，理解基于 Webpack 的工程化方案。</p>
2. 复杂网页开发	2.1 ECMAScript2015 (ES6)	<p>2.1.1 能够了解 ES6 和 ES5 的区别，了解 E6+ 的相关拓展。</p> <p>2.1.2 能够掌握 ES6 引入的新特性，掌握 ES6 的核心语法。</p> <p>2.1.3 能够掌握 ES6 Promise 异步编程。</p> <p>2.1.4 能够理解 ES6 Module 的关键点。</p>
	2.2 TypeScript 应用	<p>2.2.1 能够理解 JavaScript 的不足，理解 TypeScript 产生的背景和优点。</p> <p>2.2.2 能够搭建 TypeScript 开发环境，能使用第三方库简化流程。</p> <p>2.2.3 能够理解 TypeScript 中的各种静态类型，掌握 TypeScript 语法和使用方式。</p> <p>2.2.4 能够在主流框架中整合 TypeScript，实现核心功能。</p>
	2.3 主流框架应用	<p>2.3.1 能够了解常见的前端主流框架的种类，特点和优缺点。</p> <p>2.3.2 能够了解 MVVM 模式与 MVP 模式与 MVC 模式的区别，能掌握 MVVM 原理中生命周期、虚拟 DOM、内容渲染等。</p> <p>2.3.3 能够至少熟练掌握一个主流前端框架，理解相关实现原理，熟悉其语法和核心 API。</p> <p>2.3.4 能够配合主流打包、编译、构建工具，掌握并优化组合使用。</p>
3. 移动端网页开发	3.1 移动端基础	<p>3.1.1 能够了解移动端的概念以及与 PC 端的区别。</p> <p>3.1.2 能够了解移动端屏幕、移动端浏览器、操作系统的不同。</p> <p>3.1.3 能够了解主流移动端页面开发技术选型与常用适配方案。</p> <p>3.1.4 能够了解和解决移动端固定定位、字体单位选择、光标颜色等常见问题。</p> <p>3.1.5 能够解决主流前端框架应用中 PC 端和移动端页面开发中的适配和兼容性问题。</p>
	3.2 响应式	<p>3.2.1 能够掌握 HTML5 多媒体技术在移动端页面上的应用。</p> <p>3.2.2 能够掌握响应式布局开发与 Flex 布局，掌握 CSS 响应式方案。</p>

工作领域	工作任务	职业技能要求
		<p>3.2.3 能够通过网页布局、样式和业务逻辑处理，实现适配不同终端的页面。</p> <p>3.2.4 能够掌握媒体查询中的媒体类型以及媒体特性的概念与用法，能使用媒体查询实现响应式布局效果。</p> <p>3.2.5 能够熟悉主流前端框架中响应式设计及其原理，并能根据业务需求实现相关功能。</p>
	3.3 小程序开发	<p>3.3.1 能够了解小程序的概念、开发环境搭建和配置。</p> <p>3.3.2 能够掌握小程序基础组件与操作。</p> <p>3.3.3 能够掌握小程序常用 API。</p> <p>3.3.4 能够运用主流前端框架开发小程序应用的相关功能。</p> <p>3.3.5 能够综合应用网页开发技巧和小程序基础知识，独立开发小程序。</p>

表 3 Web 全栈开发职业技能等级要求（高级）

工作领域	工作任务	职业技能要求
1. 服务器端开发	1.1 Node.js 应用	<p>1.1.1 能够理解服务端开发和前端开发思路上的区别，了解 Node.js 产生的背景和特点。</p> <p>1.1.2 能够理解 Node.js 事件驱动机制，掌握常见模块的使用，掌握 Node.js 生态工具的运用。</p> <p>1.1.3 能够掌握至少一种主流 Node Server 框架的使用，理解中间件实现原理。</p> <p>1.1.4 能够理解 RESTful 架构风格，能使用 Node.js 或相关框架设计和开发 RESTful API。</p> <p>1.1.5 能够理解 GraphQL 的设计理念，能够掌握 GraphQL 的 Schema 等语法，能够使用 Apollo 等框架搭建 GraphQL 服务端与客户端。</p>
	1.2 数据库应用	<p>1.2.1 能够了解主流关系型数据库和 NoSQL 数据库，及其特点和应用场景。</p> <p>1.2.2 能够掌握关系型数据库常用命令以及在 Node 中的应用。</p> <p>1.2.3 能够掌握常用 NoSQL 数据库常用命令以及在 Node 中的应用。</p> <p>1.2.4 能够掌握数据驱动和 ORM，能定义和同步模型，能针对模型完成增删查改。</p>
	1.3 Linux 与应用部署	<p>1.3.1 能够掌握基本的 Linux 服务器操作命令，能够进行基础的 Linux 环境下服务管理操作。</p> <p>1.3.2 能够理解 Docker/Kubernetes 中的基础概念，能够编写简单的 Dockerfile 文件并针对前端应用进行打包与部署。</p> <p>1.3.3 能够理解 DevOps 中的基础概念，能够结合 Gitlab</p>

工作领域	工作任务	职业技能要求
		CI/Jarvis 等工具搭建 CI/CD 工作流，能够执行线上应用的滚动部署/蓝绿部署等操作。
2. 架构设计	2.1 设计模式	<p>2.1.1 能够理解前端设计模式的核心思想和基本理念，掌握其抽象的设计原则。</p> <p>2.1.2 能够了解常用的设计模式在主流前端框架中的应用。</p> <p>2.1.3 能够理解前端应用场景中创建型模式、结构型模式和行为型模式等核心设计模式的场景和应用。</p> <p>2.1.4 能够掌握常用代码优化的语言特性，利用设计模式进行代码优化。</p>
	2.2 算法和数据结构	<p>2.2.1 能够对算法、数据结构有所了解。</p> <p>2.2.2 能够掌握如数组、链表、树等常用数据结构的特点和实现。</p> <p>2.2.3 能够理解和掌握递归（深度、广度搜索）、分治策略（如二分查找法）、排序等常用算法。</p> <p>2.2.4 能够结合前端实际开发情景，在业务中运用数据结构与算法。</p>
	2.3 框架/库的编写	<p>2.3.1 能够掌握主流前端框架的设计理念，能够分析主流前端框架的代码组织与模块构成，能够根据团队需求个性化改造/复刻相关框架。</p> <p>2.3.2 能够针对团队的业务需求，构建脚手架工具，能够统一团队内部的开发环境、代码审查、部署交付等基础流程。</p> <p>2.3.3 能够了解低代码搭建的理念与实践，能够快速构建团队物料库，能够进行领域低代码搭建平台的开发。</p>
	2.4 微前端与跨端/混合开发	<p>2.4.1 能够掌握微前端的设计理念与实现方式，能够利用 Webpack 等脚手架搭建团队内微服务框架与流程。</p> <p>2.4.2 能够了解 Electron、Cordova、React Native 等混合开发框架的使用方式，能够利用这些框架开发桌面端、移动端等应用。</p> <p>2.4.3 能够了解常见的小程序跨端框架的设计理念与实现方式，能够自行设计、开发小程序框架。</p>
3. 线上业务支撑	3.1 安全预防	<p>3.1.1 能够理解和掌握前端常见的 XSS、CSRF、Cookie 等安全问题等的预防和解决。</p> <p>3.1.2 能够了解常见 HTTP 协议的安全问题和危害，掌握常见的数据加密和鉴权方案。</p> <p>3.1.3 能够理解和掌握服务器端常见的密码安全、信息泄露等信息安全问题及其预防和解决。</p> <p>3.1.4 能够理解和掌握服务器端 SQL 注入、上传漏洞等信息安全问题的预防和解决。</p>
	3.2 性能优化	<p>3.2.1 能够深刻理解输入 URL 到页面加载完成的全流程，理解前端性能优化的切入点。</p> <p>3.2.2 能够掌握性能检测的常用工具和指标，进行性能问题的</p>

工作领域	工作任务	职业技能要求
		分析和定位。 3.2.3 能够从网络、渲染、存储、端能力等方面优化和解决性能相关问题。 3.2.4 能够利用构建工具对各种前端资源加载进行优化。
	3.3 用户体验中台	3.3.1 能够掌握 APM 的核心关注指标与追踪维度，能够针对团队业务个性化定制体验度量指标。 3.3.2 能够了解常见网页分析统计、追踪反馈平台的部署与使用方式，能够快速为线上业务集成埋点统计分析的能力。 3.3.3 能够掌握前端错误捕获、上报与追踪的方法，能够搭建错误追踪平台。能够了解用户行为捕获与复现的方式，能够为用户提供实时反馈能力。

参考文献

- [1] GB-T5271.1-2000 信息技术 词汇 第1部分：基本术语
- [2] GB/T 18792-2002 信息技术 文件描述和处理语言 超文本置标语言(HTML)
- [3] GB_T 37668-2019 信息技术 互联网内容无障碍可访问性技术要求与测试方法
- [4] 中华人民共和国职业分类大典（2015年版）
- [5] 中等职业学校专业目录（2010年修订）
- [6] 普通高等学校高等职业教育（专科）专业目录（2015年）
- [7] 普通高等学校本科专业目录（2020年版）
- [8] 中等职业学校专业教学标准（试行）
- [9] 高等职业学校专业教学标准（2018年）
- [10] 高等职业学校软件技术专业顶岗实习标准
- [11] 国家职业教育改革实施方案（2019年）
- [12] 中华人民共和国高等教育法（2018年修正本）