

选择题

**一、Git 选择题 (25道) **

1. 初始化本地Git仓库的命令是？ ()
A. `git init` B. `git start` C. `git create` D. `git new`
2. 查看工作区文件状态的命令是？ ()
A. `git log` B. `git status` C. `git diff` D. `git check`
3. 将工作区修改添加到暂存区的命令是？ ()
A. `git commit` B. `git push` C. `git add` D. `git stage`
4. 提交暂存区到本地仓库的命令是？ ()
A. `git commit -m "message"` B. `git push -m "message"`
C. `git save -m "message"` D. `git log -m "message"`
5. 将本地提交推送到远程仓库的命令是？ ()
A. `git fetch` B. `git pull` C. `git push` D. `git send`
6. 从远程仓库拉取最新代码的命令是？ ()
A. `git push` B. `git pull` C. `git fetch` D. `git get`
7. 查看提交历史日志的命令是？ ()
A. `git status` B. `git log` C. `git history` D. `git show`
8. 创建新分支的命令是？ ()
A. `git branch 分支名` B. `git new branch 分支名`
C. `git create branch 分支名` D. `git make branch 分支名`
9. 切换到指定分支的命令是？ ()
A. `git switch 分支名` B. `git go 分支名`
C. `git change 分支名` D. `git move 分支名`
10. 合并指定分支到当前分支的命令是？ ()
A. `git merge 分支名` B. `git combine 分支名`
C. `git join 分支名` D. `git mix 分支名`
11. 删除本地分支的命令是？ ()

- A. `git delete 分支名` B. `git branch -d 分支名`
C. `git remove 分支名` D. `git rm 分支名`

12. 放弃工作区所有修改的命令是? ()

- A. `git reset` B. `git checkout .` C. `git clean` D. `git undo`

13. 查看两次提交差异的命令是? ()

- A. `git diff 提交1 提交2` B. `git compare 提交1 提交2`
C. `git difftool 提交1 提交2` D. `git check 提交1 提交2`

14. 暂存当前工作区修改, 切换到其他分支的命令是? ()

- A. `git save` B. `git stash` C. `git hold` D. `git temp`

15. 恢复暂存的修改的命令是? ()

- A. `git stash pop` B. `git stash apply` C. `git stash restore` D. `git stash get`

16. 克隆远程仓库到本地的命令是? ()

- A. `git copy 远程地址` B. `git clone 远程地址`
C. `git download 远程地址` D. `git get 远程地址`

17. 查看本地所有分支的命令是? ()

- A. `git branch` B. `git branches` C. `git list branch` D. `git show branch`

18. 查看远程分支的命令是? ()

- A. `git remote branch` B. `git branch -r` C. `git show remote` D. `git list remote`

19. 为远程仓库设置别名的命令是? ()

- A. `git remote add 别名 远程地址` B. `git remote set 别名 远程地址`
C. `git remote alias 别名 远程地址` D. `git remote name 别名 远程地址`

20. 撤销最近一次提交, 但保留修改的命令是? ()

- A. `git commit --undo` B. `git reset --soft HEAD~1`
C. `git revert HEAD` D. `git undo last`

21. 强制推送本地分支到远程 (覆盖远程) 的命令是? ()

- A. `git push -f` B. `git push --force` C. 以上都对 D. 以上都不对

22. `git fetch` 与 `git pull` 的区别是? ()
- A. `fetch` 拉取远程分支但不合并, `pull` 拉取并合并
 - B. `fetch` 只拉取代码, `pull` 拉取并提交
 - C. 两者无区别
 - D. `fetch` 用于本地, `pull` 用于远程
23. 标记某个提交为里程碑的命令是? ()
- A. `git tag 版本号`
 - B. `git mark 版本号`
 - C. `git release 版本号`
 - D. `git version 版本号`
24. 查看标签列表的命令是? ()
- A. `git tag`
 - B. `git list tag`
 - C. `git show tag`
 - D. `git tags`
25. 推送标签到远程仓库的命令是? ()
- A. `git push origin 标签名`
 - B. `git push tag 标签名`
 - C. `git push --tag 标签名`
 - D. `git send tag 标签名`

**二、HTML 选择题 (25道) **

1. HTML5 文档的正确声明是? ()
- A. `<!DOCTYPE html5>`
 - B. `<!DOCTYPE html>`
 - C. `<html5>`
 - D. `<html doctype="html5">`
2. 下列哪个是 HTML5 新增的语义化标签? ()
- A. `<div>`
 - B. `<span>`
 - C. `<article>`
 - D. `<font>`
3. 用于定义页面头部的标签是? ()
- A. `<header>`
 - B. `<head>`
 - C. `<top>`
 - D. `<heading>`
4. 用于定义页面底部的标签是? ()
- A. `<bottom>`
 - B. `<footer>`
 - C. `<foot>`
 - D. `<end>`
5. 用于创建超链接的标签是? ()
- A. `<link>`
 - B. `<a>`
 - C. `<href>`
 - D. `<url>`
6. `<a>` 标签中, 用于指定链接地址的属性是? ()
- A. `src`
 - B. `link`
 - C. `href`
 - D. `url`
7. 用于插入图片的标签是? ()

A. `` B. `` C. `` D. ``

8. `` 标签中, 用于指定图片路径的属性是? ()

A. `href` B. `src` C. `path` D. `url`

9. 用于创建无序列表的标签是? ()

A. `

` B. `

` C. `

` D. ` - `

10. 用于创建有序列表的标签是? ()

A. `

` B. `

` C. `

` D. ` - `

11. 表格中, 用于定义表头单元格的标签是? ()

A. ` ` B. ` ` C. `` D. `` | |

12. 表格中, 用于定义表格行的标签是? ()

A. `| |
| --- |
|` B. `` C. `` D. ` ` |

13. HTML5 中, 用于播放视频的标签是? ()

A. `` B. `` C. `` D. ``

14. HTML5 中, 用于播放音频的标签是? ()

A. `` B. `` C. `` D. ``

15. 表单中, 用于提交表单的按钮类型是? ()

A. `type="button"` B. `type="submit"` C. `type="reset"` D. `type="send"`

16. 表单中, 用于重置表单的按钮类型是? ()

A. `type="button"` B. `type="submit"` C. `type="reset"` D. `type="clear"`

17. HTML5 中, 用于输入邮箱的表单类型是? ()

A. `type="email"` B. `type="mail"` C. `type="e-mail"` D. `type="emailaddress"`

18. HTML5 中, 用于输入数字的表单类型是? ()

A. `type="number"` B. `type="digit"` C. `type="num"` D. `type="integer"`

19. 用于定义文档主体的标签是? ()

A. `` B. `` C. `` D. ``

20. ``<meta>`` 标签中, 用于设置字符编码的属性是? ()

A. ``charset`` B. ``encoding`` C. ``code`` D. ``set``

21. 用于定义文档标题的标签是? ()

A. ``<title>`` B. ``<h1>`` C. ``<header>`` D. ``<caption>``

22. HTML5 中, 用于绘制图形的标签是? ()

A. ``<graph>`` B. ``<draw>`` C. ``<canvas>`` D. ``<svg>``

23. 用于定义短引用的标签是? ()

A. ``<quote>`` B. ``<q>`` C. ``<cite>`` D. ``<ref>``

24. 用于定义长引用的标签是? ()

A. ``<blockquote>`` B. ``<longquote>`` C. ``<cite>`` D. ``<ref>``

25. HTML5 中, 用于定义导航的标签是? ()

A. ``<nav>`` B. ``<navigation>`` C. ``<menu>`` D. ``<navigate>``

**三、CSS 选择题 (25道) **

1. 用于设置元素背景颜色的 CSS 属性是? ()

A. ``color`` B. ``bgcolor`` C. ``background-color`` D. ``background``

2. 用于设置字体大小的 CSS 属性是? ()

A. ``font-size`` B. ``text-size`` C. ``font-weight`` D. ``text-style``

3. 用于设置字体粗细的 CSS 属性是? ()

A. ``font-bold`` B. ``font-weight`` C. ``text-bold`` D. ``text-weight``

4. 用于设置文本水平对齐方式的 CSS 属性是? ()

A. ``text-align`` B. ``align-text`` C. ``text-position`` D. ``align``

5. 用于清除浮动的常用方法是? ()

A. ``clear: both`` B. ``float: none`` C. ``clear: all`` D. ``float: clear``

6. CSS 中, 类选择器以什么符号开头? ()

A. `#` B. `` C. `\*` D. `:`

7. CSS 中, ID 选择器以什么符号开头? ()

A. `#` B. `` C. `\*` D. `:`

8. 用于设置元素内边距的 CSS 属性是? ()

A. `margin` B. `padding` C. `border` D. `spacing`

9. 用于设置元素外边距的 CSS 属性是? ()

A. `margin` B. `padding` C. `border` D. `spacing`

10. CSS 盒模型从内到外的顺序是? ()

- A. 内容区 → 边框 → 内边距 → 外边距
- B. 内容区 → 内边距 → 边框 → 外边距
- C. 内边距 → 内容区 → 边框 → 外边距
- D. 外边距 → 边框 → 内边距 → 内容区

11. `box-sizing: border-box` 的作用是? ()

- A. 元素宽度包含边框和内边距
- B. 元素宽度仅包含内容区
- C. 清除元素默认边距
- D. 设置元素为盒子模型

12. 用于设置元素边框的 CSS 属性是? ()

A. `border` B. `border-style` C. `border-width` D. `border-color`

13. Flex 布局中, 用于设置主轴方向对齐方式的属性是? ()

A. `align-items` B. `justify-content` C. `flex-direction` D. `align-content`

14. Flex 布局中, 用于设置交叉轴方向对齐方式的属性是? ()

A. `align-items` B. `justify-content` C. `flex-direction` D. `align-content`

15. 用于设置元素圆角的 CSS 属性是? ()

A. `border-radius` B. `border-round` C. `corner` D. `radius`

16. 用于设置元素阴影的 CSS 属性是? ()

A. `shadow` B. `box-shadow` C. `element-shadow` D. `drop-shadow`

17. CSS3 中, 用于实现过渡动画的属性是? ()
A. `animation` B. `transition` C. `transform` D. `keyframe`
18. CSS3 中, 用于定义动画关键帧的规则是? ()
A. `@animation` B. `@keyframe` C. `@keyframes` D. `@transition`
19. 用于将元素隐藏且不占空间的 CSS 属性是? ()
A. `visibility: hidden` B. `display: none` C. `opacity: 0` D. `hidden: true`
20. 用于设置元素透明度的 CSS 属性是? ()
A. `transparent` B. `opacity` C. `alpha` D. `transparency`
21. 媒体查询中, 用于匹配屏幕宽度小于等于768px的写法是? ()
A. `@media (max-width: 768px)` B. `@media (min-width: 768px)`
C. `@media (width <= 768px)` D. `@media screen and (max-width: 768px)`
22. 用于设置元素浮动的 CSS 属性是? ()
A. `float` B. `align` C. `position` D. `display`
23. 用于设置元素定位方式为绝对定位的属性值是? ()
A. `position: relative` B. `position: absolute`
C. `position: fixed` D. `position: static`
24. Grid 布局中, 用于定义列数和列宽的属性是? ()
A. `grid-template-rows` B. `grid-template-columns`
C. `grid-columns` D. `grid-rows`
25. 用于设置文本装饰 (如下划线) 的 CSS 属性是? ()
A. `text-decoration` B. `text-style` C. `font-decoration` D. `text-line`

**四、Vue3 选择题 (25道) **

1. Vue3 中, 组合式 API 的入口函数是? ()
A. `created()` B. `setup()` C. `mounted()` D. `init()`
2. 用于创建响应式对象的 API 是? ()

- A. ``ref()`` B. ``reactive()`` C. ``ref`` 和 ``reactive`` D. ``reactiveRef()``
3. ``ref`` 定义的响应式数据，访问时需要通过什么属性？（ ）
A. ``value`` B. ``data`` C. ``get()`` D. 直接访问
4. Vue3 中，用于监听数据变化的 API 是？（ ）
A. ``watch()`` B. ``watchEffect()`` C. 以上都对 D. ``onChange()``
5. 用于在模板中渲染数据的语法是？（ ）
A. ``{{ 数据 }}`` B. ``{ 数据 }`` C. ``[[ 数据 ]]` D. ``<% 数据 %>``
6. ``v-bind`` 指令的缩写是？（ ）
A. ``:`` B. ``@`` C. ``#`` D. ``$``
7. ``v-on`` 指令的缩写是？（ ）
A. ``:`` B. ``@`` C. ``#`` D. ``$``
8. 用于条件渲染的指令是？（ ）
A. ``v-for`` B. ``v-if`` C. ``v-show`` D. ``v-if`` 和 ``v-show``
9. ``v-if`` 和 ``v-show`` 的主要区别是？（ ）
A. ``v-if`` 是真实创建/销毁元素，``v-show`` 是切换 ``display`` 属性
B. ``v-if`` 性能更好 C. ``v-show`` 支持 ``<template>`` 标签 D. 无区别
10. 用于列表渲染的指令是？（ ）
A. ``v-for`` B. ``v-list`` C. ``v-loop`` D. ``v-repeat``
11. Vue3 中，组件的 ``props`` 定义方式是？（ ）
A. 在 ``props`` 选项中声明 B. 在 ``setup`` 中通过 ``defineProps`` 声明
C. 直接在模板中使用 D. 通过 ``props()`` 函数定义
12. 子组件向父组件传递数据的方式是？（ ）
A. ``props`` B. ``emit`` C. ``provide/inject`` D. ``$parent``
13. 用于跨层级组件通信的 API 是？（ ）
A. ``props/emits`` B. ``provide/inject`` C. ``$bus`` D. ``localStorage``

14. Vue3 中, 生命周期钩子函数的写法是? ()
- A. 选项式: `mounted() {}` B. 组合式: `onMounted(() => {})`
C. 以上都对 D. 以上都不对
15. `setup` 函数的执行时机是? ()
- A. 在 `beforeCreate` 之前 B. 在 `created` 之后
C. 在 `mounted` 之前 D. 在 `beforeMount` 之后
16. Vue Router 中, 用于声明路由规则的配置项是? ()
- A. `routes` B. `paths` C. `routers` D. `links`
17. 用于路由跳转的组件是? ()
- A. `<RouterLink>` B. `<Link>` C. `<RouteLink>` D. `<a>`
18. 用于渲染路由组件的出口是? ()
- A. `<RouterView>` B. `<RouteView>` C. `<RouterOutlet>` D. `<RouteOutlet>`
19. Vue3 中, 用于获取路由参数的 API 是? ()
- A. `$route.params` B. `useRoute().params` C. `useRouter().params` D. `route.params`
20. Pinia 中, 定义仓库的函数是? ()
- A. `createStore` B. `defineStore` C. `createPinia` D. `useStore`
21. Vue3 中, `computed` 的作用是? ()
- A. 定义方法 B. 定义计算属性 (缓存的派生数据)
C. 监听数据变化 D. 执行异步操作
22. 用于绑定表单元素的指令是? ()
- A. `v-model` B. `v-bind` C. `v-form` D. `v-input`
23. Vue3 中, `defineEmits` 的作用是? ()
- A. 声明组件可触发的事件 B. 定义组件的 `props`
C. 注册全局事件 D. 监听父组件事件
24. 用于动态组件切换的组件是? ()
- A. `<DynamicComponent>` B. `<Component :is="组件名">`
C. `<SwitchComponent>` D. `<ChangeComponent>`

25. Vue3 相比 Vue2 的主要变化不包括? ()
- A. 新增组合式 API
 - B. 响应式系统基于 Proxy
 - C. 移除 `v-model` 指令
 - D. 优化性能

以下是 Git、HTML、CSS、Vue3 四类选择题的完整答案，按模块整理如下：

**一、Git 选择题 (25道) **

1. A (`git init` 初始化本地仓库)
2. B (`git status` 查看工作区状态)
3. C (`git add` 将修改添加到暂存区)
4. A (`git commit -m "message"` 提交到本地仓库)
5. C (`git push` 推送本地提交到远程)
6. B (`git pull` 拉取远程最新代码)
7. B (`git log` 查看提交历史)
8. A (`git branch 分支名` 创建新分支)
9. A (`git switch 分支名` 切换分支)
10. A (`git merge 分支名` 合并分支)
11. B (`git branch -d 分支名` 删除本地分支)
12. B (`git checkout .` 放弃工作区所有修改)
13. A (`git diff 提交1 提交2` 查看两次提交差异)
14. B (`git stash` 暂存当前工作区修改)
15. A (`git stash pop` 恢复暂存的修改并删除 stash 记录)
16. B (`git clone 远程地址` 克隆远程仓库)
17. A (`git branch` 查看本地所有分支)
18. B (`git branch -r` 查看远程分支)
19. A (`git remote add 别名 远程地址` 设置远程仓库别名)
20. B (`git reset --soft HEAD~1` 撤销最近一次提交但保留修改)
21. C (`git push -f` 和 `git push --force` 均为强制推送)
22. A (`fetch` 拉取不合并, `pull` 拉取并合并)
23. A (`git tag 版本号` 标记里程碑)
24. A (`git tag` 查看标签列表)
25. A (`git push origin 标签名` 推送标签到远程)

**二、HTML 选择题 (25道) **

1. B (HTML5 文档声明为 `<!DOCTYPE html>`)
2. C (`<article>` 是 HTML5 新增语义化标签)
3. A (`<header>` 定义页面头部)
4. B (`<footer>` 定义页面底部)
5. B (`<a>` 标签创建超链接)
6. C (`<a>` 标签的 `href` 属性指定链接地址)
7. A (`<img>` 标签插入图片)
8. B (`<img>` 标签的 `src` 属性指定图片路径)
9. A (`<ul>` 创建无序列表)
10. B (`<ol>` 创建有序列表)
11. B (`<th>` 定义表头单元格)
12. A (`<tr>` 定义表格行)
13. A (`<video>` 播放视频)
14. A (`` 播放音频)
15. B (`type="submit"` 是提交按钮)
16. C (`type="reset"` 是重置按钮)
17. A (`type="email"` 用于输入邮箱)
18. A (`type="number"` 用于输入数字)
19. A (`<body>` 定义文档主体)
20. A (`<meta charset="UTF-8">` 设置字符编码)
21. A (`<title>` 定义文档标题)
22. C (`<canvas>` 用于绘制图形)
23. B (`<q>` 定义短引用)
24. A (`<blockquote>` 定义长引用)
25. A (`<nav>` 定义导航)

**三、CSS 选择题 (25道) **

1. C (`background-color` 设置背景颜色)
2. A (`font-size` 设置字体大小)
3. B (`font-weight` 设置字体粗细)
4. A (`text-align` 设置文本水平对齐)
5. A (`clear: both` 清除浮动)
6. B (类选择器以 `.` 开头)
7. A (ID 选择器以 `#` 开头)
8. B (`padding` 设置内边距)

9. A (‘margin’ 设置外边距)
10. B (盒模型顺序: 内容区 → 内边距 → 边框 → 外边距)
11. A (‘box-sizing: border-box’ 让宽度包含边框和内边距)
12. A (‘border’ 综合设置边框)
13. B (‘justify-content’ 控制主轴对齐)
14. A (‘align-items’ 控制交叉轴对齐)
15. A (‘border-radius’ 设置圆角)
16. B (‘box-shadow’ 设置元素阴影)
17. B (‘transition’ 实现过渡动画)
18. C (‘@keyframes’ 定义动画关键帧)
19. B (‘display: none’ 隐藏元素且不占空间)
20. B (‘opacity’ 设置透明度)
21. A (‘@media (max-width: 768px)’ 匹配屏幕宽度≤768px)
22. A (‘float’ 设置元素浮动)
23. B (‘position: absolute’ 是绝对定位)
24. B (‘grid-template-columns’ 定义列数和列宽)
25. A (‘text-decoration’ 设置文本装饰)

**四、Vue3 选择题 (25道) **

1. B (‘setup()’ 是组合式 API 入口)
2. C (‘ref’ 和 ‘reactive’ 均可创建响应式数据)
3. A (‘ref’ 数据需通过 ‘.value’ 访问)
4. C (‘watch()’ 和 ‘watchEffect()’ 均可监听数据变化)
5. A (‘{{ 数据 }}’ 用于模板渲染)
6. A (‘v-bind’ 缩写为 ‘:’)
7. B (‘v-on’ 缩写为 ‘@’)
8. D (‘v-if’ 和 ‘v-show’ 均用于条件渲染)
9. A (‘v-if’ 操作 DOM 存在性, ‘v-show’ 切换 ‘display’)
10. A (‘v-for’ 用于列表渲染)
11. B (Vue3 中 ‘props’ 通过 ‘defineProps’ 声明)
12. B (子组件通过 ‘emit’ 向父组件传数据)
13. B (‘provide/inject’ 用于跨层级通信)
14. C (选项式和组合式均有对应生命周期写法)
15. A (‘setup’ 在 ‘beforeCreate’ 之前执行)
16. A (‘routes’ 配置路由规则)
17. A (‘<RouterLink>’ 用于路由跳转)

- 18. A (`<RouterView>` 是路由出口)
- 19. B (`useRoute().params` 获取路由参数)
- 20. B (`defineStore` 定义 Pinia 仓库)
- 21. B (`computed` 定义计算属性)
- 22. A (`v-model` 绑定表单元素)
- 23. A (`defineEmits` 声明组件可触发的事件)
- 24. B (`<Component :is="组件名">` 实现动态组件切换)
- 25. C (Vue3 未移除 `v-model`, 反而增强了功能)

二.多选

以下是 Git、HTML、CSS、Vue3 四个模块的多选题（每个模块20道），聚焦核心知识点的多维度考察：

一、Git 多选题（20道）

- 1. 下列属于 Git 初始化仓库相关的命令有（ ）
 - A. `git init` B. `git clone` C. `git remote add` D. `git start`
- 2. 可将工作区修改添加到暂存区的命令有（ ）
 - A. `git add .` B. `git add filename` C. `git stage filename` D. `git commit -a`
- 3. 用于查看提交历史的命令有（ ）
 - A. `git log` B. `git log --oneline` C. `git history` D. `git reflog`
- 4. 下列属于分支操作的命令有（ ）
 - A. `git branch` B. `git switch` C. `git merge` D. `git tag`
- 5. 可用于撤销提交的命令有（ ）
 - A. `git reset --soft HEAD~1` B. `git revert HEAD` C. `git commit --amend` D. `git undo`
- 6. 与远程仓库交互的命令有（ ）
 - A. `git push` B. `git pull` C. `git fetch` D. `git remote`
- 7. 可暂存工作区修改的命令有（ ）
 - A. `git stash` B. `git stash save` C. `git hold` D. `git temp`

8. 下列关于 `git fetch` 和 `git pull` 的说法正确的有 ()
- A. `git fetch` 拉取远程分支但不合并
 - B. `git pull` = `git fetch` + `git merge`
 - C. 两者均会修改工作区代码
 - D. `git fetch` 更安全, 适合多人协作
9. 用于删除分支的命令有 ()
- A. `git branch -d 分支名` B. `git branch -D 分支名` C. `git rm branch 分支名` D. `git delete 分支名`
10. 可查看文件差异的命令有 ()
- A. `git diff` B. `git diff --cached` C. `git diff 分支1 分支2` D. `git compare`
11. 提交代码时必须包含的信息有 ()
- A. 提交说明 (`-m` 参数) B. 暂存区的修改 C. 远程仓库地址 D. 分支名称
12. 下列属于标签相关的命令有 ()
- A. `git tag v1.0` B. `git push origin v1.0` C. `git tag -d v1.0` D. `git show v1.0`
13. 可用于解决合并冲突的操作有 ()
- A. 手动编辑冲突文件 B. `git merge --abort` C. `git rebase --continue` D. `git conflict solve`
14. 下列关于 `git reset` 和 `git revert` 的区别有 ()
- A. `git reset` 会修改提交历史 B. `git revert` 会创建新提交抵消旧提交
 - C. `git reset` 适合本地未推送的提交 D. `git revert` 适合已推送的公开提交
15. 查看本地分支的命令有 ()
- A. `git branch` B. `git branch -l` C. `git branch -r` D. `git branch --all`
16. 下列可用于创建新分支并切换的命令有 ()
- A. `git branch 分支名` & `git switch 分支名` B. `git switch -c 分支名`
 - C. `git checkout -b 分支名` D. `git new branch 分支名`
17. 下列属于 Git 工作区状态的有 ()
- A. 未跟踪 (untracked) B. 已暂存 (staged) C. 已修改 (modified) D. 已提交 (committed)
18. 强制推送的命令有 ()
- A. `git push -f` B. `git push --force` C. `git push origin 分支名 --force` D. `git push --hard`

19. 可查看远程仓库信息的命令有 ()

A. `git remote` B. `git remote -v` C. `git remote show origin` D. `git remote info`

20. 下列属于 Git 基本工作流程的步骤有 ()

A. 工作区修改 B. 暂存区提交 C. 本地仓库提交 D. 远程仓库推送

**二、HTML 多选题 (20道) **

1. 属于 HTML5 新增语义化标签的有 ()

A. `` B. `` C. `

` D. `

`

2. 下列属于块级元素的有 ()

A. `

` B. `

` C. `

` D. ``

3. 表单中可输入文本的元素有 ()

A. `

4. HTML5 新增的表单类型有 ()

A. `email` B. `tel` C. `date` D. `search`

5. 下列属于 `

A. `src` B. `alt` C. `width` D. `href`

6. 可实现列表的标签有 ()

A. `

7. 表格相关的标签有 ()

A. `

8. HTML5 中用于媒体播放的标签有 ()

A. `

9. 表单验证相关的属性有 ()

A. `required` B. `pattern` C. `min` D. `max`

10. 下列属于 `

A. `href` B. `target` C. `rel` D. `src`

11. 可用于定义文档元数据的标签有 ()

A. `` B. `` C. `<link>` D. `<style>`</p></div><div data-bbox="36 155 401 173" data-label="Text"><p>12. HTML5 中用于绘制图形的标签有 ()</p></div><div data-bbox="57 181 534 198" data-label="Text"><p>A. `<canvas>` B. `<svg>` C. `<graph>` D. `<draw>`</p></div><div data-bbox="36 230 370 248" data-label="Text"><p>13. 下列属于语义化文本标签的有 ()</p></div><div data-bbox="57 255 568 272" data-label="Text"><p>A. `<strong>` B. `<em>` C. `<span>` D. `<blockquote>`</p></div><div data-bbox="36 304 289 321" data-label="Text"><p>14. 表单中按钮的类型有 ()</p></div><div data-bbox="57 329 464 347" data-label="Text"><p>A. `submit` B. `reset` C. `button` D. `image`</p></div><div data-bbox="36 379 398 396" data-label="Text"><p>15. 下列属于 `<form>` 标签属性的有 ()</p></div><div data-bbox="57 404 493 422" data-label="Text"><p>A. `action` B. `method` C. `enctype` D. `target`</p></div><div data-bbox="36 453 341 471" data-label="Text"><p>16. HTML5 新增的全局属性有 ()</p></div><div data-bbox="57 478 594 496" data-label="Text"><p>A. `data-\*` B. `contenteditable` C. `hidden` D. `spellcheck`</p></div><div data-bbox="36 527 410 545" data-label="Text"><p>17. 可用于创建下拉列表的标签组合有 ()</p></div><div data-bbox="57 553 560 595" data-label="Text"><p>A. `<select>` + `<option>` B. `<input list>` + `<datalist>`
C. `<dropdown>` + `<item>` D. `<select>` + `<datalist>`</p></div><div data-bbox="36 627 423 645" data-label="Text"><p>18. 下列属于 `<head>` 标签内的元素有 ()</p></div><div data-bbox="57 652 498 670" data-label="Text"><p>A. `<title>` B. `<meta>` C. `<link>` D. `<script>`</p></div><div data-bbox="36 702 370 720" data-label="Text"><p>19. 用于定义分区的语义化标签有 ()</p></div><div data-bbox="57 727 534 744" data-label="Text"><p>A. `<section>` B. `<article>` C. `<aside>` D. `<div>`</p></div><div data-bbox="36 776 361 794" data-label="Text"><p>20. HTML5 中支持的存储方式有 ()</p></div><div data-bbox="57 802 658 819" data-label="Text"><p>A. `localStorage` B. `sessionStorage` C. `cookie` D. `IndexedDB`</p></div><div data-bbox="34 875 341 893" data-label="Section-Header"><h3>### **三、CSS 多选题 (20道) **</h3></div><div data-bbox="36 900 359 918" data-label="Text"><p>1. 可用于设置元素背景的属性有 ()</p></div><div data-bbox="51 925 926 944" data-label="Text"><p>A. `background-color` B. `background-image` C. `background-repeat` D. `background-position`</p></div>

2. 用于设置字体样式的属性有 ()

A. `font-family` B. `font-size` C. `font-weight` D. `font-style`

3. Flex 布局中控制主轴对齐的 `justify-content` 取值有 ()

A. `flex-start` B. `center` C. `space-between` D. `align-center`

4. 下列属于盒模型组成部分的有 ()

A. 内容区 (content) B. 内边距 (padding) C. 边框 (border) D. 外边距 (margin)

5. 可实现元素隐藏的方式有 ()

A. `display: none` B. `visibility: hidden` C. `opacity: 0` D. `position: absolute; left: -9999px`

6. CSS 选择器中优先级从高到低的有 ()

A. ID 选择器 (`#box`) B. 类选择器 (`.box`) C. 标签选择器 (`div`) D. 通配符 (`\*`)

7. 用于设置文本样式的属性有 ()

A. `text-align` B. `text-decoration` C. `line-height` D. `letter-spacing`

8. CSS3 新增的特性有 ()

A. `border-radius` B. `box-shadow` C. `flex` D. `@keyframes`

9. 媒体查询中常用的断点设置有 ()

A. `max-width: 768px` (移动端) B. `min-width: 768px` (平板)
C. `min-width: 1024px` (桌面端) D. `width: 375px` (固定宽度)

10. 用于定位的 `position` 属性取值有 ()

A. `static` B. `relative` C. `absolute` D. `fixed`

11. 可用于清除浮动的方法有 ()

A. 父元素添加 `overflow: hidden` B. 父元素添加伪元素 `::after { content: "; clear: both; display: block; }`
C. 浮动元素后添加空标签 `` D. 父元素也设置浮动

12. 用于设置元素边框的属性有 ()

A. `border-width` B. `border-style` C. `border-color` D. `border-radius`

13. CSS3 中实现动画的属性有 ()

A. `transition` B. `animation` C. `transform` D. `keyframe`

14. `transform` 属性的常用取值有 ()

A. `translate()` B. `rotate()` C. `scale()` D. `skew()`

15. 用于设置元素尺寸的属性有 ()

A. `width` B. `height` C. `min-width` D. `max-height`

16. 下列属于响应式布局方案的有 ()

A. 媒体查询 (`@media`) B. Flex 布局 C. Grid 布局 D. 固定宽度布局

17. 可实现元素水平居中的方式有 ()

A. 块级元素设置 `margin: 0 auto` B. Flex 容器设置 `justify-content: center`

C. 绝对定位元素设置 `left: 50%; transform: translateX(-50%)` D. `text-align: center` (子元素为行内元素)

18. 用于设置元素间距的属性有 ()

A. `margin` B. `padding` C. `gap` (Flex/Grid 中) D. `spacing`

19. CSS 中可继承的属性有 ()

A. `font-size` B. `color` C. `text-align` D. `border`

20. Grid 布局中定义行列的属性有 ()

A. `grid-template-columns` B. `grid-template-rows` C. `grid-column` D. `grid-row`

**四、Vue3 多选题 (20道) **

1. Vue3 组合式 API 包括 ()

A. `setup()` B. `ref()` C. `reactive()` D. `onMounted()`

2. 用于创建响应式数据的 API 有 ()

A. `ref()` B. `reactive()` C. `toRefs()` D. `computed()`

3. 下列关于 `ref` 和 `reactive` 的说法正确的有 ()

A. `ref` 可用于基本类型和对象 B. `reactive` 仅用于对象

C. `ref` 访问需加 `.value` D. `reactive` 访问直接使用

4. 用于监听数据变化的 API 有 ()

A. `watch()` B. `watchEffect()` C. `onUpdated()` D. `computed()`

5. Vue3 生命周期钩子函数包括 ()

A. `onBeforeMount()` B. `onMounted()` C. `onBeforeUpdate()` D. `onUnmounted()`

6. 组件间通信方式有 ()

A. `props` + `emit` B. `provide` + `inject` C. Pinia D. `$parent` / `$children`

7. `v-model` 可用于绑定的表单元素有 ()

A. `<input type="text">` B. `<textarea>` C. `<select>` D. `<checkbox>`

8. 条件渲染指令有 ()

A. `v-if` B. `v-else` C. `v-else-if` D. `v-show`

9. 列表渲染 `v-for` 可遍历的类型有 ()

A. 数组 B. 对象 C. 字符串 D. 数字

10. Vue Router 中用于路由跳转的方式有 ()

A. `<RouterLink to="/path">` B. `useRouter().push('/path')`

C. `this.$router.push('/path')` D. `<a href="/path">`

11. 路由配置中常用的属性有 ()

A. `path` B. `name` C. `component` D. `children`

12. Pinia 仓库的核心组成部分有 ()

A. `state` B. `getters` C. `actions` D. `mutations`

13. `computed` 的特点有 ()

A. 缓存计算结果 B. 依赖数据变化时自动更新

C. 可用于异步操作 D. 支持读写模式 (`get` / `set`)

14. Vue3 中支持的指令有 ()

A. `v-bind` B. `v-on` C. `v-for` D. `v-slot`

15. 动态组件相关的用法有 ()

- A. `<Component :is="componentName">` B. `keep-alive` 缓存组件
C. `v-if` 切换组件 D. `component()` 函数
16. 下列属于 Vue3 相比 Vue2 的改进有 ()
A. 响应式系统基于 Proxy B. 新增组合式 API
C. 移除 `v-model` D. 更好的 TypeScript 支持
17. `setup` 函数的参数有 ()
A. `props` B. `context` C. `this` D. `data`
18. 可用于组件注册的方式有 ()
A. 全局注册 (`app.component()`) B. 局部注册 (`components` 选项)
C. 自动导入 (Vite 插件) D. `import` 后直接使用
19. Vue3 中处理异步请求的方式有 ()
A. 在 `setup` 中直接使用 `async/await` B. 结合 `onMounted` 钩子
C. 使用 Pinia 的 `actions` D. 使用 `watchEffect` 监听异步结果
20. `defineProps` 和 `defineEmits` 的作用有 ()
A. `defineProps` 声明组件接收的 props
B. `defineEmits` 声明组件可触发的事件
C. 两者均无需导入即可使用
D. 用于替代 Vue2 的 `props` 和 `$emit`

以下是 Git、HTML、CSS、Vue3 四类多选题的完整答案，按模块整理如下：

**一、Git 多选题 (20道) **

1. **ABC** (`git init` 初始化本地仓库, `git clone` 克隆远程仓库, `git remote add` 关联远程仓库, 均属于初始化相关操作)
2. **ABC** (`git add .` 暂存所有修改, `git add filename` 暂存指定文件, `git stage` 是 `git add` 的别名; `git commit -a` 直接提交已跟踪文件的修改, 跳过暂存)
3. **ABD** (`git log` 查看提交历史, `git log --oneline` 简化显示, `git reflog` 查看所有操作记录; `git history` 不是 Git 命令)
4. **ABC** (`git branch` 管理分支, `git switch` 切换分支, `git merge` 合并分支; `git tag` 是标签操作)

5. **ABC** (`git reset --soft HEAD~1` 撤销最近提交并保留修改, `git revert HEAD` 创建新提交抵消旧提交, `git commit --amend` 修正最近一次提交; `git undo` 不是 Git 命令)
6. **ABCD** (`git push` 推送, `git pull` 拉取, `git fetch` 抓取, `git remote` 管理远程仓库, 均为远程交互命令)
7. **AB** (`git stash` 和 `git stash save` 均可暂存工作区修改; `git hold`、`git temp` 不是 Git 命令)
8. **ABD** (`git fetch` 仅拉取不合并, `git pull` 是拉取+合并, `fetch` 更安全, 避免直接修改工作区)
9. **AB** (`git branch -d` 安全删除已合并分支, `git branch -D` 强制删除未合并分支; C、D 不是 Git 命令)
10. **ABC** (`git diff` 查看工作区与暂存区差异, `git diff --cached` 查看暂存区与本地仓库差异, `git diff 分支1 分支2` 查看分支差异; `git compare` 不是 Git 命令)
11. **AB** (提交需包含暂存区修改和提交说明; 远程地址和分支名非必需, 默认使用当前配置)
12. **ABCD** (`git tag v1.0` 创建标签, `git push origin v1.0` 推送标签, `git tag -d v1.0` 删除本地标签, `git show v1.0` 查看标签详情)
13. **ABC** (手动编辑冲突文件解决冲突, `git merge --abort` 放弃合并, `git rebase --continue` 解决冲突后继续变基; `git conflict solve` 不是 Git 命令)
14. **ABCD** (`reset` 修改历史, 适合本地未推送提交; `revert` 创建新提交, 适合已推送的公开提交)
15. **AB** (`git branch` 和 `git branch -l` 均查看本地分支; `-r` 查看远程分支, `--all` 查看所有分支)
16. **ABC** (A 是分步创建并切换, B、C 是一步创建并切换; D 不是 Git 命令)
17. **ABCD** (Git 工作区状态包括未跟踪、已暂存、已修改、已提交)
18. **ABC** (`git push -f`、`git push --force`、`git push origin 分支名 --force` 均为强制推送; `git push -hard` 不是 Git 命令)
19. **ABC** (`git remote` 查看远程仓库别名, `git remote -v` 查看详细地址, `git remote show origin` 查看指定远程仓库信息; `git remote info` 不是 Git 命令)
20. **ABCD** (Git 基本流程: 工作区修改 → 暂存区提交 → 本地仓库提交 → 远程仓库推送)

**二、HTML 多选题 (20道) **

1. **ABC** (`<header>`、`<footer>`、`<section>` 是 HTML5 新增语义化标签; `<div>` 是传统无语义标签)
2. **ABC** (`<div>`、`<p>`、`<h1>` 是块级元素; `<span>` 是行内元素)
3. **ABC** (`<input type="text">`、`<textarea>`、`<input type="password">` 均可输入文本; `<input type="number">` 限制为数字)
4. **ABCD** (`email`、`tel`、`date`、`search` 均为 HTML5 新增表单类型)
5. **ABC** (`<img>` 标签的 `src` 设路径, `alt` 设替代文本, `width` 设宽度; `href` 是 `<a>` 标签属性)
6. **ABC** (`<ul>` 无序列表, `<ol>` 有序列表, `<dl>` 定义列表; `<list>` 不是 HTML 标签)
7. **ABCD** (`<table>` 定义表格, `<tr>` 定义行, `<td>` 定义单元格, `<th>` 定义表头)

8. **AB** (`<video>` 播放视频, `<audio>` 播放音频; `<media>` 不是标准标签, `<audio>` 是音频标签的错误写法)
9. **ABCD** (`required` 必填, `pattern` 正则验证, `min/max` 限制范围, 均用于表单验证)
10. **ABC** (`<a>` 标签的 `href` 设链接, `target` 设打开方式, `rel` 设关系; `src` 是 `<img>` 等标签属性)
11. **ABC** (`<meta>`、`<title>`、`<link>` 用于定义元数据; `<style>` 用于嵌入样式, 不属于元数据)
12. **AB** (`<canvas>` 动态绘制图形, `<svg>` 定义矢量图形; `<graph>`、`<draw>` 不是标准标签)
13. **ABD** (`<strong>` 强调重要性, `<em>` 强调语气, `<blockquote>` 长引用, 均为语义化文本标签; `<span>` 无语义)
14. **ABCD** (`submit` 提交表单, `reset` 重置表单, `button` 普通按钮, `image` 图像按钮, 均为表单按钮类型)
15. **ABCD** (`<form>` 标签的 `action` 设提交地址, `method` 设提交方式, `enctype` 设编码类型, `target` 设打开方式)
16. **ABCD** (`data-*` 自定义数据, `contenteditable` 可编辑, `hidden` 隐藏, `spellcheck` 拼写检查, 均为 HTML5 新增全局属性)
17. **AB** (`<select>` + `<option>` 创建标准下拉列表, `<input list>` + `<datalist>` 创建带提示的输入框; C、D 不是标准标签组合)
18. **ABCD** (`<title>`、`<meta>`、`<link>`、`<script>` 均需放在 `<head>` 标签内)
19. **ABC** (`<section>`、`<article>`、`<aside>` 是语义化分区标签; `<div>` 无语义)
20. **ABD** (HTML5 支持 `localStorage`、`sessionStorage`、`IndexedDB`; `cookie` 是 HTTP 标准, 非 HTML5 新增)

三、CSS 多选题 (20道)

1. **ABCD** (`background-color` 背景色, `background-image` 背景图, `background-repeat` 重复方式, `background-position` 位置, 均用于设置背景)
2. **ABCD** (`font-family` 字体类型, `font-size` 大小, `font-weight` 粗细, `font-style` 样式, 均用于字体设置)
3. **ABC** (`flex-start` 起点对齐, `center` 居中, `space-between` 两端对齐; `align-center` 不是 `justify-content` 的取值)
4. **ABCD** (盒模型从内到外: 内容区 (content) → 内边距 (padding) → 边框 (border) → 外边距 (margin))
5. **ABCD** (`display: none` 隐藏且不占空间, `visibility: hidden` 隐藏但占空间, `opacity: 0` 透明, 绝对定位偏移均能实现隐藏效果)
6. **ABCD** (优先级从高到低: ID 选择器 → 类选择器 → 标签选择器 → 通配符)
7. **ABCD** (`text-align` 水平对齐, `text-decoration` 装饰, `line-height` 行高, `letter-spacing` 字间距, 均用于文本样式)

8. ****ABCD**** (`border-radius` 圆角, `box-shadow` 阴影, `flex` 弹性布局, `@keyframes` 动画关键帧, 均为 CSS3 新增特性)
9. ****ABC**** (常用响应式断点: 移动端 ($\leq 768\text{px}$)、平板 ($\geq 768\text{px}$)、桌面端 ($\geq 1024\text{px}$); 固定宽度不适合响应式)
10. ****ABCD**** (`static` 默认定位, `relative` 相对定位, `absolute` 绝对定位, `fixed` 固定定位, 均为 `position` 取值)
11. ****ABCD**** (四种清除浮动的常用方法: 父元素设 `overflow: hidden`、父元素加伪元素、浮动后加空标签、父元素也浮动)
12. ****ABC**** (`border-width` 宽度, `border-style` 样式, `border-color` 颜色, 用于设置边框; `border-radius` 是圆角属性)
13. ****AB**** (`transition` 过渡动画, `animation` 关键帧动画; `transform` 是变换属性, `@keyframes` 是动画规则, 不是属性)
14. ****ABCD**** (`translate()` 平移, `rotate()` 旋转, `scale()` 缩放, `skew()` 倾斜, 均为 `transform` 常用取值)
15. ****ABCD**** (`width`/`height` 设宽高, `min-width`/`max-height` 设最小/最大尺寸, 均用于控制元素尺寸)
16. ****ABC**** (媒体查询、Flex、Grid 均支持响应式布局; 固定宽度布局不响应)
17. ****ABCD**** (四种常见水平居中方式: 块级元素 `margin: 0 auto`、Flex 容器 `justify-content: center`、绝对定位偏移、行内元素父容器 `text-align: center`)
18. ****ABC**** (`margin` 外边距, `padding` 内边距, `gap` 网格/弹性间距, 均用于设置间距; `spacing` 不是 CSS 属性)
19. ****ABC**** (`font-size` 字体大小, `color` 颜色, `text-align` 文本对齐, 均可继承; `border` 边框不可继承)
20. ****AB**** (`grid-template-columns` 定义列, `grid-template-rows` 定义行; `grid-column`/`grid-row` 用于定位网格项)

****四、Vue3 多选题 (20道) ****

1. ****ABCD**** (`setup()` 入口函数, `ref()`/`reactive()` 创建响应式数据, `onMounted()` 生命周期钩子, 均属于组合式 API)
2. ****ABC**** (`ref()`/`reactive()` 创建响应式数据, `toRefs()` 将响应式对象转为 refs; `computed()` 是计算属性, 不直接创建响应式数据)
3. ****ABCD**** (`ref` 支持基本类型和对象, 需通过 `.value` 访问; `reactive` 仅支持对象, 直接访问)
4. ****AB**** (`watch()` 监听指定数据, `watchEffect()` 监听依赖数据; `onUpdated()` 是更新后钩子, `computed()` 是计算属性)
5. ****ABCD**** (均为 Vue3 组合式 API 中的生命周期钩子)

6. ****ABC**** (``props``+``emit`` 父子通信, ``provide``+``inject`` 跨层级通信, Pinia 全局状态管理; ``$parent``/``$children`` 在 Vue3 中不推荐使用)
7. ****ABCD**** (``v-model`` 可绑定文本输入框、文本域、下拉框、复选框等表单元素)
8. ****ABCD**** (``v-if``/``v-else``/``v-else-if`` 条件渲染 (操作 DOM) , ``v-show`` 条件显示 (切换 ``display``))
9. ****ABCD**** (``v-for`` 可遍历数组、对象、字符串、数字)
10. ****ABC**** (``<RouterLink>`` 组件跳转, ``useRouter().push()`` 编程式跳转 (组合式) , ``this.$router.push()`` 编程式跳转 (选项式) ; ``<a>`` 标签跳转刷新页面, 不推荐)
11. ****ABCD**** (``path`` 路由路径, ``name`` 路由名称, ``component`` 对应组件, ``children`` 嵌套路由, 均为路由配置常用属性)
12. ****ABC**** (Pinia 仓库包含 ``state`` (状态)、``getters`` (计算属性)、``actions`` (方法) ; 移除了 Vuex 的 ``mutations``)
13. ****ABD**** (``computed`` 缓存结果, 依赖变化自动更新, 支持 ``get``/``set`` 读写模式; 不支持异步操作)
14. ****ABCD**** (``v-bind`` 绑定属性, ``v-on`` 绑定事件, ``v-for`` 列表渲染, ``v-slot`` 插槽, 均为 Vue3 支持的指令)
15. ****AB**** (``<Component :is>`` 动态切换组件, ``keep-alive`` 缓存组件; ``v-if`` 切换组件不是动态组件语法, ``component()`` 不是 Vue 方法)
16. ****ABD**** (Vue3 响应式基于 Proxy, 新增组合式 API, 更好支持 TypeScript; 未移除 ``v-model``, 反而增强)
17. ****AB**** (``setup`` 函数接收 ``props`` (组件参数) 和 ``context`` (上下文对象) ; 无 ``this`` 和 ``data`` 参数)
18. ****ABC**** (全局注册、局部注册、自动导入均为组件注册方式; ``import`` 后需注册才能使用)
19. ****ABCD**** (Vue3 中可在 ``setup`` 用 ``async/await``、结合生命周期钩子、Pinia 的 ``actions``、``watchEffect`` 处理异步请求)
20. ****ABCD**** (``defineProps`` 声明接收的 ``props``, ``defineEmits`` 声明触发的事件, 两者无需导入, 替代 Vue2 的 ``props`` 和 ``$emit``)

以上答案覆盖了各模块核心知识点的多维度考察, 可用于检验对 Git、HTML、CSS、Vue3 综合应用的掌握程度。

填空多选

以下是 Git、HTML、CSS、Vue3 四个模块的填空题 (共40道) 和简答题 (共20道) , 聚焦核心概念和实战应用:

****一、填空题 (每个模块10道) ****

Git 填空题

1. 初始化本地 Git 仓库的命令是 _____。
2. 将工作区所有修改添加到暂存区的命令是 _____。
3. 提交暂存区到本地仓库并添加说明的命令是 _____。
4. 查看工作区与暂存区差异的命令是 _____。
5. 从远程仓库拉取最新代码并合并到当前分支的命令是 _____。
6. 创建并切换到新分支的命令是 _____（使用 `switch`）。
7. 暂存当前工作区修改的命令是 _____。
8. 查看本地所有分支的命令是 _____。
9. 强制推送本地分支到远程的命令是 _____。
10. 撤销最近一次提交但保留修改的命令是 _____。

HTML 填空题

1. HTML5 文档的声明标签是 _____。
2. 用于定义页面主要内容的语义化标签是 _____。
3. `` 标签中，用于指定图片替代文本的属性是 _____。
4. HTML5 中用于播放视频的标签是 _____。
5. 表单中，用于提交表单的按钮类型是 _____。
6. 定义有序列表的标签是 _____。
7. `` 标签中，用于在新窗口打开链接的 `target` 属性值是 _____。
8. HTML5 中用于输入日期的表单类型是 _____。
9. 用于定义文档元数据的标签是 _____。
10. 表格中，用于合并单元格的属性是 _____（横向合并）。

CSS 填空题

1. 用于清除浮动的 CSS 属性及值是 _____。
2. Flex 布局中，控制子元素在交叉轴居中的属性是 _____。
3. `box-sizing: border-box` 的作用是 _____。
4. 用于设置元素圆角的 CSS 属性是 _____。
5. 媒体查询中，匹配屏幕宽度小于等于 768px 的写法是 _____。
6. 定位方式中，元素相对于自身初始位置偏移的是 _____。
7. 用于设置元素透明度的 CSS 属性是 _____。
8. CSS 中，类选择器的语法是 _____（例：选择类名为 `box` 的元素）。
9. 实现元素水平垂直居中的 Flex 布局父容器属性组合是 _____。
10. 用于定义动画关键帧的 CSS 规则是 _____。

Vue3 填空题

1. 组合式 API 的入口函数是 _____。
2. 用于创建基本类型响应式数据的 API 是 _____。
3. `v-bind` 指令的缩写是 _____。
4. 监听数据变化的 API 中，自动追踪依赖的是 _____。
5. 组件间通过 `props` 接收父组件数据时，在组合式 API 中需使用 _____ 声明。
6. Vue Router 中，用于渲染路由组件的出口标签是 _____。
7. Pinia 中定义仓库的函数是 _____。
8. `v-model` 在复选框上绑定的值类型是 _____。
9. 生命周期钩子中，组件挂载完成后执行的是 _____。
10. 动态组件切换使用的标签是 _____，配合 `:is` 属性。

二、简答题（每个模块5道）

Git 简答题

1. 简述 `git pull` 和 `git fetch` 的区别，以及各自的适用场景。
2. 如何在 Git 中创建一个标签并推送到远程仓库？
3. 当合并分支发生冲突时，解决冲突的步骤是什么？
4. 什么是 `git stash`？使用场景有哪些？
5. 简述 Git 工作区、暂存区、本地仓库、远程仓库的关系。

HTML 简答题

1. 简述 HTML5 语义化标签的优势，并列举3个常用语义化标签及其作用。
2. 表单中 `required` 和 `pattern` 属性的作用是什么？举例说明。
3. `` 标签的作用是什么？有几种关联表单元素的方式？
4. HTML5 新增了哪些表单类型？至少列举4种并说明用途。
5. 简述 `` 和 `` 的区别。

CSS 简答题

1. 简述 CSS 盒模型的两种模式（`content-box` 和 `border-box`）的区别。
2. Flex 布局中 `flex-direction` 的取值有哪些？分别代表什么含义？
3. 如何实现一个元素在父容器中水平垂直居中？至少写出3种方法。

4. 简述 `display: none`、`visibility: hidden`、`opacity: 0` 的区别。
5. 什么是响应式布局？实现响应式布局的常用方法有哪些？

Vue3 简答题

1. 简述 Vue3 中 `ref` 和 `reactive` 的区别及使用场景。
2. `v-if` 和 `v-show` 的区别是什么？分别适用于什么场景？
3. 组件间通信的方式有哪些？请分别说明适用场景。
4. 简述 Vue Router 的基本使用流程（从安装到路由跳转）。
5. 什么是 Pinia？它相比 Vuex 有哪些改进？

参考答案

填空题答案

Git

1. `git init`
2. `git add .`
3. `git commit -m "提交说明"`
4. `git diff`
5. `git pull`
6. `git switch -c 分支名`
7. `git stash`
8. `git branch`
9. `git push -f`（或 `git push --force`）
10. `git reset --soft HEAD~1`

HTML

1. `<!DOCTYPE html>`
2. `<main>`
3. `alt`
4. `<video>`
5. `submit`
6. `<ol>`
7. `\_blank`
8. `date`
9. `<meta>`

10. `colspan`

****CSS****

1. `clear: both`
2. `align-items: center`
3. 元素宽度包含边框和内边距
4. `border-radius`
5. `@media (max-width: 768px)`
6. `position: relative`
7. `opacity`
8. `.box`
9. `display: flex; justify-content: center; align-items: center`
10. `@keyframes`

****Vue3****

1. `setup()`
2. `ref()`
3. `:`
4. `watchEffect()`
5. `defineProps`
6. ``
7. `defineStore`
8. 布尔值 (`boolean`) 或数组 (`array`)
9. `onMounted`
10. ``

**简答题答案要点**

****Git****

1. - `git fetch`：拉取远程分支到本地但不合并，适合查看远程变更后再手动合并，更安全。
- `git pull`：拉取并自动合并到当前分支，适合确认远程变更可直接合并的场景。
2. 步骤：
- 创建标签：`git tag v1.0`（轻量标签）或 `git tag -a v1.0 -m "版本说明"`（附注标签）。
- 推送标签：`git push origin v1.0`。
3. 步骤：

- 打开冲突文件，手动编辑解决冲突（保留需要的代码，删除冲突标记 `<<<<<<` 等）。
- 标记为已解决：`git add 冲突文件`。
- 完成合并：`git commit -m "解决冲突"`。

4. `git stash` 用于暂存当前工作区未提交的修改（工作区恢复干净状态）。

场景：需要切换分支但不想提交当前修改、拉取远程代码前暂存本地修改等。

5. 关系：

- 工作区：本地编辑的文件区域。
- 暂存区：临时存放待提交的修改（`git add` 后）。
- 本地仓库：存储提交历史（`git commit` 后）。
- 远程仓库：远程服务器上的仓库（`git push` 同步本地提交）。

****HTML****

1. 优势：提升代码可读性、SEO 友好、便于屏幕阅读器解析。

标签：`<header>`（页头）、`<nav>`（导航）、`<article>`（独立内容）。

2. - `required`：标记字段为必填项，提交时自动验证。

- `pattern`：指定正则表达式验证输入格式，例：`<input type="text" pattern="\d{6}" title="6位数字">`。

3. 作用：关联表单元素，扩大点击区域，提升可访问性。

关联方式：

- `for` 属性 + 元素 `id`：`<label for="name">姓名</label><input id="name">`。
- 嵌套元素：`<label>姓名：<input></label>`。

4. 新增类型：

- `email`：验证邮箱格式。
- `tel`：适合输入电话号码（移动端调出数字键盘）。
- `date`：日期选择器。
- `range`：滑块控件，用于选择范围值。

5. 区别：

- `<canvas>`：基于像素的动态绘图，需通过 JS 绘制，不支持事件交互（需手动实现）。
- `<svg>`：矢量图形，基于 XML 标签定义，支持事件交互，缩放不失真。

****CSS****

1. 区别:

- `content-box` (默认): 宽度 = 内容区宽度 (不包含 padding 和 border)。
- `border-box`: 宽度 = 内容区 + padding + border (便于布局计算)。

2. `flex-direction` 取值:

- `row` (默认): 主轴水平, 从左到右。
- `row-reverse`: 主轴水平, 从右到左。
- `column`: 主轴垂直, 从上到下。
- `column-reverse`: 主轴垂直, 从下到上。

3. 方法:

- Flex 布局: 父容器 `display: flex; justify-content: center; align-items: center`。
- 定位 + transform: 子元素 `position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%)`。
- Grid 布局: 父容器 `display: grid; place-items: center`。

4. 区别:

- `display: none`: 元素消失, 不占空间, 不触发事件。
- `visibility: hidden`: 元素隐藏, 仍占空间, 不触发事件。
- `opacity: 0`: 元素透明, 仍占空间, 可触发事件。

5. 响应式布局: 使页面在不同设备上自适应显示。

方法: 媒体查询 (`@media`)、Flex/Grid 布局、相对单位 (`%`、`rem`、`vw`)。

****Vue3****

1. 区别:

- `ref`: 用于基本类型 (Number、String 等) 和对象, 访问需加 `.value`, 适合简单值。
 - `reactive`: 仅用于对象/数组, 直接访问属性, 适合复杂对象。
- 场景: 简单数据用 `ref`, 复杂状态 (如用户信息对象) 用 `reactive`。

2. 区别:

- `v-if`: 动态创建/销毁元素 (操作 DOM), 初始渲染开销低, 切换开销高。
 - `v-show`: 切换 `display: none` (不操作 DOM), 初始渲染开销高, 切换开销低。
- 场景: `v-if` 适合条件少变 (如权限判断), `v-show` 适合频繁切换 (如标签页)。

3. 通信方式：

- ``props`` + ``emit``：父子组件通信（父传子用 ``props``，子传父用 ``emit``）。
- ``provide`` + ``inject``：跨层级组件通信（祖先传后代）。
- Pinia：全局状态管理（多组件共享数据）。
- 事件总线（``mitt``）：任意组件通信（适合简单场景）。

4. Vue Router 流程：

- 安装：``npm install vue-router``。
- 定义路由组件：创建页面组件。
- 配置路由规则：``const routes = [{ path: '/', component: Home }]``。
- 创建路由实例：``const router = createRouter({ routes, history: createWebHistory() })``。
- 挂载路由：``app.use(router)``。
- 跳转：使用 ``<RouterLink to="/">`` 或 ``useRouter().push('/')``。

5. Pinia 是 Vue 的状态管理库，改进：

- 移除 ``mutations``，直接在 ``actions`` 中修改状态。
- 更好的 TypeScript 支持。
- 简化 API，无需嵌套模块。
- 支持组合式 API 写法。

Vue3

以下是专门针对 Vue3 的单选题（20道）、多选题（15道）、填空题（10道）和简答题（8道），聚焦 Vue3 核心特性与实战应用：

一、Vue3 单选题（20道）

1. Vue3 中，组合式 API 的核心入口函数是？（ ）

- A. ``created()`` B. ``setup()`` C. ``mounted()`` D. ``init()``

2. 用于创建基本类型响应式数据的 API 是？（ ）

- A. ``reactive()`` B. ``ref()`` C. ``toRefs()`` D. ``computed()``

3. ``ref`` 定义的响应式数据，在模板中访问时需要加 ``value`` 吗？（ ）

- A. 需要 B. 不需要 C. 仅对象类型需要 D. 仅数组类型需要

4. Vue3 中, `v-model` 在自定义组件上的默认绑定事件是? ()
A. `input` B. `change` C. `update:modelValue` D. `modelChange`
5. 下列哪个 API 可以自动追踪依赖并执行副作用? ()
A. `watch()` B. `watchEffect()` C. `onMounted()` D. `computed()`
6. 组合式 API 中, 组件挂载完成后执行的生命周期钩子是? ()
A. `onCreated()` B. `onMounted()` C. `onBeforeMount()` D. `onUpdated()`
7. 用于将 `reactive` 对象转换为 `ref` 对象的 API 是? ()
A. `toRef()` B. `toRefs()` C. `ref()` D. `isRef()`
8. Vue3 中, `setup` 函数的返回值可以是? ()
A. 一个对象 (供模板使用) B. 一个渲染函数 C. 以上都可以 D. 只能是对象
9. 下列哪个不是 Vue3 移除的 API? ()
A. `\$on` B. `\$off` C. `\$set` D. `\$emit`
10. Vue Router 4 中, 用于获取当前路由信息的 API 是? ()
A. `useRoute()` B. `useRouter()` C. `\$route` D. `\$router`
11. Pinia 中, 修改状态的方式是? ()
A. 通过 `mutations` B. 直接在 `actions` 中修改 C. 通过 `commit` D. 通过 `dispatch`
12. `v-for` 遍历数组时, 推荐绑定的 `key` 属性值是? ()
A. 数组索引 B. 唯一 ID C. 数组元素 D. 随机数
13. 组合式 API 中, 声明组件接收的 `props` 需使用? ()
A. `props` 选项 B. `defineProps()` C. `props()` D. `useProps()`
14. 下列哪个指令用于动态绑定 HTML 特性? ()
A. `v-on` B. `v-bind` C. `v-model` D. `v-text`
15. Vue3 响应式系统的底层实现基于? ()
A. `Object.defineProperty` B. `Proxy` C. `Reflect` D. `Promise`
16. 用于缓存组件状态的标签是? ()

A. ``<keep-alive>`` B. ``<cache>`` C. ``<component-cache>`` D. ``<preserve>``

17. ``computed`` 属性的特点是? ()

A. 不缓存结果 B. 依赖变化时自动更新 C. 支持异步操作 D. 必须返回基本类型

18. 下列哪个不是 Vue3 新增的特性? ()

A. 组合式 API B. 多根节点组件 C. 过滤器 (`filter`) D. ``<script setup>``

19. ``<script setup>`` 语法中, 导入的组件需要注册吗? ()

A. 需要 B. 不需要, 自动注册 C. 仅全局组件需要 D. 仅局部组件需要

20. 用于声明组件可触发事件的 API 是? ()

A. ``defineEmits()`` B. ``emit()`` C. ``$emit()`` D. ``useEmits()``

**二、Vue3 多选题 (15道) **

1. Vue3 组合式 API 包含哪些核心函数? ()

A. ``ref()`` B. ``reactive()`` C. ``watch()`` D. ``onMounted()``

2. 下列关于 ``ref`` 和 ``reactive`` 的说法正确的有? ()

A. ``ref`` 可用于基本类型和对象 B. ``reactive`` 仅支持对象和数组
C. ``ref`` 对象访问需加 ``.value`` D. ``reactive`` 对象解构后仍保持响应式

3. 可用于监听数据变化的 API 有? ()

A. ``watch()`` B. ``watchEffect()`` C. ``onUpdated()`` D. ``computed()``

4. Vue3 支持的生命周期钩子有? ()

A. ``onBeforeMount()`` B. ``onMounted()`` C. ``onBeforeUnmount()`` D. ``onUnmounted()``

5. 组件间通信的方式包括? ()

A. ``props`` + ``emit`` B. ``provide`` + ``inject`` C. Pinia D. 事件总线 (mitt)

6. ``v-model`` 可绑定的表单元素有? ()

A. 文本输入框 B. 复选框 C. 下拉框 D. 单选按钮

7. Vue Router 中, 路由跳转的方式有? ()

A. ``<RouterLink to="/home">`` B. ``router.push('/home')`

C. `router.replace('/home')` D. `<a href="/home">`

8. Pinia 仓库的组成部分有? ()

A. `state` B. `getters` C. `actions` D. `mutations`

9. `<script setup>` 的优势有? ()

A. 更少的模板代码 B. 组件自动注册 C. 更好的 TypeScript 支持 D. 直接使用组合式 API

10. 动态组件的用法包括? ()

A. `<Component :is="componentName" />` B. 结合 `<keep-alive>` 缓存
C. 使用 `v-if` 切换组件 D. 通过 `component()` 函数创建

11. Vue3 相比 Vue2 的改进有? ()

A. 响应式系统基于 Proxy B. 支持多根节点组件
C. 移除 `v-model` D. 优化虚拟 DOM

12. `computed` 属性支持的特性有? ()

A. 缓存计算结果 B. 依赖变化自动更新 C. 读写模式 (`get`/`set`) D. 异步计算

13. 下列属于 Vue3 全局 API 变化的有? ()

A. `Vue.createApp()` 替代 `new Vue()` B. 全局方法需通过 `app` 实例调用
C. 移除 `Vue.filter()` D. 保留 `Vue.directive()`

14. 可用于条件渲染的指令有? ()

A. `v-if` B. `v-else` C. `v-else-if` D. `v-show`

15. 组合式 API 中, `setup` 函数的参数包括? ()

A. `props` B. `context` C. `this` D. `slots`

**三、Vue3 填空题 (10道) **

1. Vue3 中, 组合式 API 的入口函数是 _____。
2. 用于创建对象类型响应式数据的 API 是 _____。
3. `watchEffect` 与 `watch` 的主要区别是 _____。
4. Vue Router 中, 用于定义路由规则的配置项是 _____。
5. Pinia 中, 获取仓库实例的函数是 _____。
6. `<script setup>` 中, 声明 `props` 的函数是 _____。

7. 生命周期钩子中，组件更新前执行的是 _____。
8. ``v-bind`` 的缩写是 _____，``v-on`` 的缩写是 _____。
9. 动态组件切换时，用于指定组件的属性是 _____。
10. Vue3 中，多根节点组件需要添加 _____ 属性避免警告。

**四、Vue3 简答题 (8道) **

1. 详细说明 Vue3 组合式 API 与 Vue2 选项式 API 的区别，以及组合式 API 的优势。
2. 简述 ``ref``、``reactive``、``toRefs`` 的使用场景和注意事项。
3. ``watch`` 和 ``watchEffect`` 的区别是什么？分别适用于什么场景？
4. 如何在 Vue3 中实现组件间通信？请列举至少4种方式并说明适用场景。
5. 简述 Vue Router 中动态路由、嵌套路由和程式化导航的实现方式。
6. Pinia 相比 Vuex 有哪些核心改进？如何在 Vue3 中使用 Pinia 管理状态？
7. ``<script setup>`` 语法糖带来了哪些便利？与普通 ``setup`` 函数有何区别？
8. 说明 Vue3 响应式系统的实现原理（基于 Proxy），相比 Vue2 的 ``Object.defineProperty`` 有哪些优势？

参考答案

单选题答案

1. B 2. B 3. B 4. C 5. B 6. B 7. B 8. C 9. D 10. A
11. B 12. B 13. B 14. B 15. B 16. A 17. B 18. C 19. B 20. A

多选题答案

1. ABCD 2. ABC 3. AB 4. ABCD 5. ABCD 6. ABCD 7. ABC 8. ABC
9. ABCD 10. AB 11. ABD 12. ABC 13. ABCD 14. ABCD 15. AB

填空题答案

1. ``setup()``
2. ``reactive()``
3. ``watchEffect`` 自动追踪依赖，``watch`` 需要显式指定监听源
4. ``routes``
5. ``useStore()``
6. ``defineProps()``

7. ``onBeforeUpdate()``
8. ``:``; ``@``
9. ``is``
10. ``key``

简答题答案要点

1. **组合式 API vs 选项式 API**

- 区别：选项式 API 按 ``data``、``methods``、``computed`` 等选项组织代码；组合式 API 按逻辑功能组织代码，通过 ``setup`` 集中编写。
- 优势：更好的代码组织（尤其大型组件）、逻辑复用更灵活（无需 mixin）、TypeScript 支持更友好、按需引入 API 减小打包体积。

2. **``ref``、``reactive``、``toRefs`` 场景与注意事项**

- ``ref``：用于基本类型（如 ``let count = ref(0)``）或单值对象，注意 JS 中需通过 ``.value`` 访问，模板中无需。
- ``reactive``：用于复杂对象/数组（如 ``const user = reactive({ name: 'foo' })``），注意不能直接赋值（会丢失响应式），需修改属性。
- ``toRefs``：将 ``reactive`` 对象转为 ``ref`` 集合（如 ``const { name } = toRefs(user)``），解决解构后响应式丢失问题。

3. **``watch`` vs ``watchEffect``**

- 区别：``watch`` 需显式指定监听源（如 ``watch(count, (newVal) => {})``），可获取新旧值，支持深度监听和立即执行；``watchEffect`` 自动追踪函数内依赖（如 ``watchEffect(() => console.log(count.value))``），无法直接获取旧值，默认立即执行。
- 场景：``watch`` 适合监听指定数据且需旧值的场景；``watchEffect`` 适合依赖动态变化的场景（如根据多个数据计算结果）。

4. **组件通信方式**

- ``props`` + ``emit``：父子组件通信（父传子用 ``props``，子传父用 ``emit`` 触发事件）。
- ``provide`` + ``inject``：跨层级通信（祖先组件 ``provide`` 提供数据，后代组件 ``inject`` 注入）。
- Pinia：全局状态管理（多组件共享数据，如用户信息、购物车）。
- 事件总线（mitt）：任意组件通信（适合简单场景，如非父子组件通知）。
- ``v-model``：父子组件双向绑定（语法糖，简化 ``props`` + ``emit``）。

5. **Vue Router 核心功能**

- 动态路由：路径中带参数，如 ``path: '/user/:id``，通过 ``route.params.id`` 获取。

- 嵌套路由：在路由配置中通过 `children` 嵌套子路由，需在父组件中添加 `` 作为出口。
- 编程式导航：使用 `router.push('/path')` 跳转，`router.replace('/path')` 替换历史，`router.go(-1)` 后退。

6. **Pinia 改进与使用**

- 改进：移除 `mutations`，直接在 `actions` 中修改状态；支持组合式 API；更好的 TypeScript 集成；无需模块嵌套。
- 使用：
 1. 安装：`npm install pinia`。
 2. 创建仓库：`const useStore = defineStore('id', { state: () => ({ count: 0 }), actions: { increment() { this.count++ } } })`。
 3. 在组件中使用：`const store = useStore(); store.increment()`。

7. **`<script setup>` 优势与区别**

- 优势：无需手动返回变量/方法（直接在模板使用）；组件导入后自动注册；简化组合式 API 写法（无需 `setup` 函数）；支持顶层 `await`。
- 区别：普通 `setup` 需返回对象供模板使用；`<script setup>` 是语法糖，编译后自动处理返回逻辑，代码更简洁。

8. **Vue3 响应式原理与优势**

- 原理：基于 `Proxy` 拦截对象的读取、设置、删除等操作，配合 `Reflect` 反射实现响应式。
- 优势：
 - 支持监听对象新增属性、删除属性（Vue2 需 `\$set` / `\$delete`）。
 - 支持监听数组索引和 `length` 变化（Vue2 不支持）。
 - 嵌套对象自动递归响应式（无需深度遍历）。
 - 性能更好，拦截粒度更细。

以上题目覆盖 Vue3 核心概念（响应式、生命周期、组件通信）、生态工具（Vue Router、Pinia）及实战场景，可针对性检验 Vue3 掌握程度。