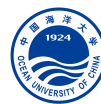


海纳百川
取则行远



中国海洋大学
OCEAN UNIVERSITY OF CHINA

Week1

左昊天

2024-08-29

Github 仓库地址

1 Latex

1.1 章节

章节: `\section{}`

子章节: `\subsection{}`

三级章节: `\subsubsection{}`

1.2 加粗

`\textbf{}`

1.3 斜体 italic

`\textit{}`

注: 中文的斜体和英文的斜体样式不一样

例: 中文样式 *English version*

1.4 下划线

`\underline{}`

1.5 添加图片及其标题

1.5.1 添加图片

先引入 `graphicx` 包, 再用 `\includegraphics{此处填图片的地址}`

1.5.2 添加标题

先将图片放在 `figure` 环境中, 再在图片的下面加 `\caption{}` 例:



图 1: 举例

遇到的问题:

(1) 添加图片后, 不显示图片

原因: 图片过大导致无法显示

解决办法：通过`\includegraphics[width=0.2\textwidth]{photo.jpg}`等比例设置图片的宽高。

(2) 图片出现在了章节前面。

原因：figure 是浮动体，系统会自动决定浮动体的放置位置

解决办法：加入宏包`\usepackage{float}`，并在`\begin{figure}`后加上 [H] 属性，强制控制浮动位置。即`\begin{figure}[H]`

1.6 改变文字颜色

加入宏包`\usepackage{xcolor}`，使用 `\textcolor{color}{想输入的文字}` 控制颜色。

1.7 列表

1.7.1 无序列表

代码：

```
1 \begin{itemize}
2   \item
3   \item
4 \end{itemize}
```

例：

- 项 1
- 项 2

1.7.2 有序列表

代码：

```
1 \begin{enumerate}
2   \item
3   \item
4 \end{enumerate}
```

例：

1. 项 1
2. 项 2

1.8 表格

```
1 \begin{tabular}{|p{3cm}|c|c|}  
2 \hline  
3 单元格1 & 单元格2 & 单元格3 \\  
4 \hline \hline  
5 单元格4 & 单元格5 & 单元格6 \\  
6 \hline  
7 单元格7 & 单元格8 & 单元格9 \\  
8 \hline  
9 \end{tabular}
```

单元格 1	单元格 2	单元格 3
单元格 4	单元格 5	单元格 6
单元格 7	单元格 8	单元格 9

解释：

1.`p{3cm}`设置列宽

2.`\hline`加横线

3.`|p{3cm}c|c|` 中的 `c` 表示该列居中对齐 (centering)

1.9 公式

1.9.1 行内公式

代码：\$此处填写公式\$

例： $F = ma$, $E = mc^2$

1.9.2 单独一行的公式

有两种写法：

写法一：

```
1 \begin{equation}  
2 此处填写公式  
3 \end{equation}
```

例：

$$F = ma \tag{1}$$

写法二：

```
1      \[
2      此处填写公式
3      \]
```

例:

$$E = mc^2$$

1.10 如何在 latex 中展示代码

1.10.1 行内代码

```
1      \verb|此处填入代码|
```

1.10.2 代码块

先引入 listings 宏包

```
\begin{lstlisting}
```

此处填写代码

```
\end{lstlisting}
```

2 Git

2.1 查看文件状态

2.1.1 文件状态

Untrack	Unmodified	Modified	Staged
未跟踪	未修改	已修改	已暂存

2.1.2 查看文件状态

`git status 文件名` 用于查看单个文件的状态

`git status` 可查看当前文件夹所有文件的状态

也可使用通配符, 如: `git status *.tex`

```
$ git status "git add.png"
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  git add.png

$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  new file:   week1/photo.jpg
  new file:   week1/week1.tex
```

2.2 添加和提交文件

2.2.1 将文件添加到暂存区

`git add 文件名` 添加单个文件

`git add .` 可将当前文件夹的所有文件添加到暂存区

也可使用通配符

补充: `git rm --cached 文件名` 可以将文件取消暂存

```
19355@Mingw64:~/Desktop/大二暑假/系统开发工具基础/tool_class_2024_sum (main)
$ git add .

19355@Mingw64:~/Desktop/大二暑假/系统开发工具基础/tool_class_2024_sum (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  new file:   week1/photo.jpg
  new file:   week1/week1.tex
```



```
19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/tool_class_2024_sum/w
eek1 (main)
$ git log --oneline
740fe98 (HEAD -> main, origin/main, origin/HEAD) 添加git的状态和提交操作
c60ead0 修改了1 latex中的内容
0cc3c96 调整了报告的风格
d6838d3 添加gitignore
481951d 完成了week1第一个内容 Latex
a4bab3b Initial commit
```

2.4 远程仓库的相关操作

2.4.1 拉取

`git pull` 将远程仓库的内容下载到本地仓库中

2.4.2 推送

`git push` 将本地仓库的内容上传到远程仓库

```
19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/tool_class_2024_sum/w
ek1 (main)
$ git push
Warning: Permanently added '[ssh.github.com]:443' (ED25519) to the list of known
hosts.
Enter passphrase for key '/c/Users/19355/.ssh/github':
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 32 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 566.09 KiB | 1.34 MiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:eric041224/tool_class_2024_sum.git
c60ead0..740fe98 main -> main
```

2.5 查询所有操作的信息

`git reflog`

```
$ git reflog
740fe98 (HEAD -> main) HEAD@{0}: reset: moving to 740fe98
0685da3 (origin/main, origin/HEAD) HEAD@{1}: commit: 查看提交信息
740fe98 (HEAD -> main) HEAD@{2}: commit: 添加git的状态和提交操作
c60ead0 HEAD@{3}: commit: 修改了1 latex中的内容
0cc3c96 HEAD@{4}: commit: 调整了报告的风格
d6838d3 HEAD@{5}: commit: 添加gitignore
481951d HEAD@{6}: commit: 完成了week1第一个内容 Latex
a4bab3b HEAD@{7}: clone: from github.com:eric041224/tool_class_2024_sum.git
```

可利用此功能撤销任意操作

2.6 版本回退

版本回退有三种:

类型	工作区	暂存区	备注
git reset --soft	√	√	回到指定版本，但此版本之后的文件依然存在，只是没提交
git reset --hard	×	×	回到指定版本，此版本后的所有文件全部被删除
git reset --mixed	√	×	与 soft 类型的区别在于需要重新将文件添加到暂存区

其中，√ 表示此区域的文件不会被删除，× 表示此区域的文件会被删除。
并且类型后面需要加上版本 ID。

```
$ git reset --soft 740fe98
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   commit1.png
    new file:   git log.png
    new file:   git log2.png
    new file:   push.png
    modified:   week1.pdf
    modified:   week1.tex

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   week1.pdf
    modified:   week1.tex

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    commit2-1.png
    commit2-2.png
```

2.7 查看差异

git diff 默认展示工作区和暂存区的差异

git diff HEAD 展示工作区与仓库的差异

git diff --cached 展示暂存区与仓库的差异

git diff 版本ID1 版本ID2 展示两个版本之间的差异

注：在这些命令后加上文件名，那么就只展示这个文件的差异

2.8 删除文件

将暂存的文件从工作区删除后，需要 git add 的命令

也可使用 git rm 直接同时删除工作区和暂存区的文件

但必须提交才能删除仓库的文件

```
$ git rm 用于测试删除命令.txt
rm 'week1/用于测试删除命令.txt'
```

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    "\347\224\250\344\272\216\346\265\213\350\257\225\345\210\240\351\231\244\345\221\275\344\273\244.txt"

19355@MingW64:~/Desktop/大二暑假/系统开发工具基础/tool_class_2024_sum/week1 (main)
$ git commit
[main 6ddfb52] 删除测试文件
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 "week1/\347\224\250\344\272\216\346\265\213\350\257\225\345\210\240\351\231\244\345\221\275\344\273\244.txt"

19355@MingW64:~/Desktop/大二暑假/系统开发工具基础/tool_class_2024_sum/week1 (main)
$ git push
Warning: Permanently added 'ssh.github.com:443' (ED25519) to the list of known hosts.
Enter passphrase for key '/c/Users/19355/.ssh/github':
Enumerating objects: 5, done.
```

2.9 分支的创建与删除

2.9.1 创建

`git branch 分支名`

2.9.2 删除

`+git branch -d 分支名` 已经合并时用这个删除

`git branch -D 分支名` 没有合并时用这个强制删除

```
$ git branch -D test
Deleted branch test (was 6ddfb52).

19355@MingW64:~/Desktop/大二暑假/系统开发工具基础/tool_class_2024_sum/week1 (main)
$ git branch
* main
```

2.10 分支的查看与切换

2.10.1 查看

`git branch`

```
$ git branch
* main
test
```

2.10.2 切换

`git switch 分支名`

```
1 (main)
$ git switch test
Switched to branch 'test'
M       week1/week1.pdf
M       week1/week1.tex
```

2.11 分支的合并

`git merge 分支名` 该命令是将指定的分支合并到当前所在的分支上。git 会自动进行合并，但当两个分支产生冲突时，需要自己手动合并。

```
19355@MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week1 (main)
$ git branch test
fatal: a branch named 'test' already exists

19355@MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week1 (main)
$ git branch -D test
Deleted branch test (was 27bb6c3).

19355@MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week1 (main)
$ git branch test

19355@MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week1 (main)
$ git switch test
Switched to branch 'test'

19355@MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week1 (test)
$ echo "该文件由分支提交" >test.txt

19355@MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week1 (test)
$ git add .
warning: in the working copy of 'week1/test.txt', LF will be replaced by CRLF the next time Git touches it
```

```
19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum
/week1 (test)
$ git commit -m "test:1"
[test 8d49b2c] test:1
1 file changed, 1 insertion(+)
create mode 100644 week1/test.txt

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum
/week1 (test)
$ git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum
/week1 (main)
$ echo "该文件由main分支创建" > mainTest.txt

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum
/week1 (main)
$ git add .
warning: in the working copy of 'week1/mainTest.txt', LF will be replaced by CRLF
the next time Git touches it

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum
/week1 (main)
$ git commit -m "main:1"
[main c8994ed] main:1
1 file changed, 1 insertion(+)
create mode 100644 week1/mainTest.txt

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum
/week1 (main)
$ git merge test
Merge made by the 'ort' strategy.
week1/test.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 week1/test.txt

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum
/week1 (main)
$ git log --graph --oneline --decorate --akk
fatal: unrecognized argument: --akk

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum
/week1 (main)
$ git log --graph --oneline --decorate --all
* a8d5153 (HEAD -> main) Merge branch 'test'
|
| * 8d49b2c (test) test:1
| * c8994ed main:1
|/
```