**.gitignore**

**Робота через термінал:**

1. **touch** .gitignore – **створення файлу**
2. **nano -w** .gitignore -**відкриваємо файл для запису**
3. **cat** newFile deletingFile– **видаляє файл і вставляє замість нього новий**
4. **mkdir** newFile – **створення нового файлу**
5. **cd** newDirectory – **перехід по дерикторіям**
6. **Git check-ignore -v** path – перевірка умісності ігнору певного шляху

**Файли з ігнор-шаблонами:**

1. <projcet>/…/.gitignore
2. <projcet>/.git/info/exclude – не являється частиною служебних деректорій і не відправляється git
3. Config: core.excludesFile
   1. Git config --global core.excludesFile ~/.gitignore
   2. XDG\_CONFIG\_HOME/git/ignore
   3. ~/.config/git/ignore

**Синтаксис ігнорування файлів без врахування шляху до них:**

1. \*.formatName({txt, md…}) – **ігнорувати ВСІ файли** із даним розширенням
2. \*.py[co] – всі файли з розширенням .pyc I .pyo, але не .pyco, так як [] – означає, що підставляється лише по одному символу вписаному в квадратні душки
3. \*.py? - ?-ігнорується конструкція + будь-який символ, окрім самої конструкції ( \*.py ) -
4. Migrate\_201[5-7] – діапазон симолів, які зарахуються до конструкції і будуть проігноровані

**Ігнорування папок:**

1. Build/ - **всі папки** з такими назвами (не файли)
2. /build/ - **ігнорується корінна папк**а build, решта папок build(script/build) не ігноруються
3. Secret/key – **проігнорований файл або дерикторія**, що знаходиться **по даному шляху (але не** doc/secret/key**)**
   1. Різниці між **/secret/key I secret/key** для gitignore немає
4. Doc/\*.html – **ігнорування html лише в даній деректорії**, але не всі
5. /\*.txt – **ігноруються файла лише в корінній деректорії**
6. var/www\*/tmp – ігнорування файлу по даній деректорії з будь-якою варіацією імені другої деректорії, якщо вона починаєтсья з www:
   1. var/www/tmp
   2. var/www-home/tmp
   3. **Але не** var/www/home/tmp – це вже порушення шляху

**\* - ! одна зірочка міняє лише частину або все ім’я але не допускає додаткової вложеності**

**\*\* - шлях до файлу чи папки можу бути довільним**

**/ - !якщо десь окрім кінця в файлу стоїть слеш – то пошук виконуєтсья з кореня проекту**

**! – знімає ігнорування з усього, що ми до цього ігнорували**

1. User/\*/private – **ігнорує файли\деректорії, які знахожяться по даному шляху**, з будь-якою назвою другої деректорії
   1. Users/john/private
   2. Users/vadym/private
   3. **Але не** users/john/project/private or script/users/vadym/private
2. \*\*/\*.log (\*.log)– **ігнорує файли не зважаючи на шлях до них**
3. \*\*/app/cache – **знаходить дану деректорію, де б вона не була в проекті**
   1. App/cache – **шукатиму в корені проекту**
   2. \*/app/cache – **шукатиме в деректорії рівно на рівень нижче від кореня проекту**
4. Docs/\*\*/\*.html **– ігноруються html файли будь-де в деректорії docs, в незалежності від кількості деректорій між ними**

**! якщо git ігнорує деректорію, він ігнорує, все, що в ній**

**Налаштування ігнорування всього, окрім чогось:**

.\* - ігноруємо усі типи файлів

!.gitignore

!.gittatributes - окрім цих файлів

**Ігноруємо всю деректорію, окрім packages.xml**

**/install/\*** - *ігнорування не самої деректорії, а усіх файлів в ній, так гіт знає, які файли є в ній і можу зробити виключення*

**!install/packages.xml**

**Неробочий варіант:**

**/install/** *- ігноруєтсья сама деректорія, гіт не знає, які файли розташовані в ній*

**!install/package.xml**