JSON

4. Создать внешний репозиторий c названием JSON.

С*оздаём новый репозиторий в самом GH.*

5. Клонировать репозиторий JSON на локальный компьютер.

К*опируем ссылку на репозиторий из GH командой git clone.*

6. Внутри локального JSON создать файл “new.json”.

touch new.json

7. Добавить файл под гит.

git add new.json

8. Закоммитить файл.

git commit –m “new json file”

9. Отправить файл на внешний GitHub репозиторий.

git push

10. Отредактировать содержание файла “new.json” - написать информацию о себе (ФИО, возраст, количество домашних животных, будущая желаемая зарплата). Всё написать в формате JSON.

cat > new.json

{

"name": "Polina",

"last name": "Dolotenkova",

"patronymic": "Olegovna",

"age": 22,

"amount\_of\_pets": 2,

"salary": 65000

}

11. Отправить изменения на внешний репозиторий.

git commit new.json -m "json update" – git push

12. Создать файл preferences.json.

touch preferences.json

13. В файл preferences.json добавить информацию о своих предпочтениях (Любимый фильм, любимый сериал, любимая еда, любимое время года, сторона которую хотели бы посетить) в формате JSON.

cat > preferences.json

{

"favourite\_film": "kill your darlings",

"favourite\_show": "black mirror",

"favourite\_food": "dried berries",

"favourite\_season": "autumn",

"country\_to\_visit": "Italy"

}

14. Создать файл sklls.json добавить информацию о скиллах которые будут изучены на курсе в формате JSON

сat >> skills.json

{

"1": "Git Bash",

"2": "HTML",

"3": "Positive testing",

"4": "Negative testing",

"5": "Smoke testing",

"6": "Regression testing"

}

15. Отправить сразу 2 файла на внешний репозиторий.

git add preferences.json

git add skills.json

git commit –m “json file”

git push

16. На веб интерфейсе создать файл bug\_report.json.

Не работает кнопка коммит.

17. Сделать Commit changes (сохранить) изменения на веб интерфейсе.

-

18. На веб интерфейсе модифицировать файл bug\_report.json, добавить баг репорт в формате JSON.

-

19. Сделать Commit changes (сохранить) изменения на веб интерфейсе.

-

20. Синхронизировать внешний и локальный репозиторий JSON.

-

XML

21. Создать внешний репозиторий c названием XML.

С*оздаём новый репозиторий в самом GH.*

22. Клонировать репозиторий XML на локальный компьютер.

К*опируем ссылку на репозиторий из GH командой git clone.*

23. Внутри локального XML создать файл “new.xml”.

touch new.xml

24. Добавить файл под гит.

git add new.xml

25. Закоммитить файл.

git commit -m "new xml"

26. Отправить файл на внешний GitHub репозиторий.

git push

27. Отредактировать содержание файла “new.xml” - написать информацию о себе (ФИО, возраст, количество домашних животных, будущая желаемая зарплата). Всё написать в формате XML

cat > new.xml

name: Polly

last name: Dolotenkova

partonymic: Olegovna

age: 22

pets: 2

salary: 65 000

28. Отправить изменения на внешний репозиторий.

git add new.xml

git commit –m “new xml”

git push

29. Создать файл preferences.xml

touch preferences.xml

30. В файл preferences.xml добавить информацию о своих предпочтениях (Любимый фильм, любимый сериал, любимая еда, любимое время года, сторона которую хотели бы посетить) в формате XML.

cat > preferences.xml

Fav movie: Madagaskar

Fav series: Euphoria

Fav food: Lollipops

Fav season: Winter

Country to visit: Qwebec

31. Создать файл sklls.xml добавить информацию о скиллах которые будут изучены на курсе в формате XML

cat >> skills.xml

1: Git Bash

2: HTML

3: Positive Testing

4: Negative testing

5: Smoke testing

6: Regression testing

32. Сделать коммит в одну строку.

git add preferences.xml | git add skills.xml | git commit -m "new xmls"

33. Отправить сразу 2 файла на внешний репозиторий.

git push

34. На веб интерфейсе создать файл bug\_report.xml.

Не работает коммит.

35. Сделать Commit changes (сохранить) изменения на веб интерфейсе.

36. На веб интерфейсе модифицировать файл bug\_report.xml, добавить баг репорт в формате XML.

-

37. Сделать Commit changes (сохранить) изменения на веб интерфейсе.

-

38. Синхронизировать внешний и локальный репозиторий XML.

-

TXT

1. Создать внешний репозиторий c названием TXT.

С*оздаём новый репозиторий в самом GH.*

2. Клонировать репозиторий TXT на локальный компьютер.

К*опируем ссылку на репозиторий из GH командой git clone.*

3. Внутри локального TXT создать файл “new.txt”.

touch new.txt

4. Добавить файл под гит.

git add new.txt

5. Закоммитить файл.

git commit -m "new txt file"

6. Отправить файл на внешний GitHub репозиторий.

git push

7. Отредактировать содержание файла “new.txt” - написать информацию о себе (ФИО, возраст, количество домашних животных, будущая желаемая зарплата). Всё написать в формате TXT.

cat > new.txt

name: Polly

last name: Dolotenkova

partonymic: Olegovna

age: 22

pets: 2

salary: 65 000

8. Отправить изменения на внешний репозиторий.

git add new.txt | git commit -m "info" | git push

9. Создать файл preferences.txt

10. В файл preferences.txt” добавить информацию о своих предпочтениях (Любимый фильм, любимый сериал, любимая еда, любимое время года, сторона которую хотели бы посетить) в формате TXT.

cat >> preferences.txt

Fav movie: Madagaskar

Fav series: Euphoria

Fav food: Lollipops

Fav season: Winter

Country to visit: Qwebec

11. Создать файл sklls.txt добавить информацию о скиллах которые будут изучены на курсе в формате TXT.

cat >> skills.txt

1: Git Bash

2: HTML

3: Positive Testing

4: Negative testing

5: Smoke testing

6: Regression testing

12. Сделать коммит в одну строку.

13. Отправить сразу 2 файла на внешний репозиторий.

git add preferences.txt | git add skills.txt | git commit -m "new files" | git push

14. На веб интерфейсе создать файл bug\_report.txt.

Не работает коммит.

15. Сделать Commit changes (сохранить) изменения на веб интерфейсе.

-

16. На веб интерфейсе модифицировать файл bug\_report.txt, добавить баг репорт в формате TXT.

-

17. Сделать Commit changes (сохранить) изменения на веб интерфейсе.

-

18. Синхронизировать внешний и локальный репозиторий TXT.

-