

## Урок 2. Знакомство с интерфейсом RStudio

Как мы уже убедились, при запуске RStudio открывается рабочая область. Она состоит из нескольких окон. Главное из них — консоль, в которую мы вводим код. Как и Python, язык R является интерпретируемым. Это означает, что мы можем писать код кусочками и сразу его выполнять (в отличие от компилируемых языков, где нельзя сразу запустить то, что мы разберем прямо сейчас). Итак, поработаем с консолью. Попробуем ввести простые строки кода и исполнить их. Начнем с вычислений:

```
2 + 4
```

```
## [1] 6
```

Чтобы исполнить код, нажмем *Enter*. Тут же в консоли отобразится результат — число 6. Попробуем что-нибудь еще:

```
3 * 8 - 4 * 5
```

```
## [1] 4
```

Или даже со скобками:

```
(3 + 5) / 2
```

```
## [1] 4
```

Теперь сохраним результат — создадим переменную. О переменной можно думать как о контейнере, в котором сохранен некоторый объект: число, строка, список, файл, таблица с данными. Создадим переменную `income` (доход) и присвоим ей значение 30:

```
income <- 30
```

Для присваивания значений в R используется специфический оператор, стрелочка, которая состоит из знака «больше» и дефиса. Технически можно использовать и обычный знак равенства:

```
income = 30
```

Все работает, но это считается некорректным, «не-аровским» способом присваивания значений. Так лучше не делать, и в официальных тьюториалах и документации вы такого не увидите.

В окне справа, во вкладке *Environment* (среда) отображаются все переменные, которые мы создаем в RStudio. Вот мы создали переменную `income` и она отображается здесь — указано ее название и значение 30.

Добавим нечисловую переменную `month` (месяц) и присвоим ей значение строковое значение `август`:

```
month <- "август"
```

Как и в Python, в R текст вводится в кавычках, или в двойных, или в одинарных. Теперь во вкладке *Environment* у нас две переменные. Подробнее о переменных мы поговорим в последующих уроках, а сейчас посмотрим на оставшееся окно внизу.

Во вкладке *Files* перечислены файлы и папки, которые находятся в рабочей директории, рабочей папке по умолчанию, той, из которой запускается RStudio. Во вкладке *Plots* отображаются графики (отсюда их можно выгрузить в файл, есть кнопка *Export*). Во вкладке *Packages* перечислены установленные библиотеки (о них поговорим позднее), а во вкладке *Help* перечислены ссылки на официальные ресурсы и документацию. Помощь также можно запросить по конкретной функции, конкретной операции. Например, рассмотрим функцию `print()` и выведем на экран значение переменной `month`:

```
print(month)
```

```
## [1] "август"
```

Функция `print()` выводит значение или сообщение на экран. А как получить более развернутое описание? Запросить помощь, поставив перед названием функции знак вопроса:

```
?print
```

Страница с документацией выводится в правом нижнем окне. Посмотреть результат можно [здесь](#). Что мы видим в документации?

- общее описание функции (раздел *Description*) — что она делает;
- использование функции (раздел *Usage*) — какие аргументы или опции есть у функции;
- пояснения по каждому аргументу (раздел *Arguments*);
- технические детали (раздел *Details*), обычно ориентированы более продвинутых пользователей;
- ссылки на статьи или книги (раздел *References*), в которых описывается функция, что особенно актуально для «содержательных» функций, используемых для построения статистических моделей — какие алгоритмы и поправки задействованы;
- ссылки на документацию по связанным функциям (раздел *See Also*);
- примеры использования функции (раздел *Examples*), к сожалению, часто требуют достаточно уверенных знаний R, чтобы их понять.

Или, если это более привычнее, можно использовать `help()`, так же как в Python или в других языках программирования:

```
help(print)
```

Результат получится тот же, что и выше.

После этого урока вам предлагается выполнить небольшое практическое задание. Чтобы избежать проблем, на всякий случай поясню, как понять, что что-то в консоли пошло не так. Когда R готов к работе, в строке консоли слева видна стрелочка `>`. Если мы ввели код некорректно, забыли скобку или что-то еще, R зависнет, ожидая окончания кода и слева будет стоять плюсики `+`. В таких случаях можно либо нажать *Esc*, и R выйдет из режима ожидания, либо, если известно, в чем именно ошибка, дописать недостающий символ прямо в новой строке.

Теперь приступайте к первому практическому заданию. Желаю удачи!