

Посвящается моим Учителям:

Моему отцу Ивану Владимировичу Павлову;
инженеру Виктору Григорьевичу Струц;
первому директору ФизМатШколы
Михаилу Алексеевичу Могилевскому;
старшему оперуполномоченному
Дмитрию Юрьевичу Пучкову.

Отдаю свой Долг перед Вами:

Сначала вы учили меня.

Теперь я учу других.

Такая вот связь поколений.

Низкий Вам поклон!

P. S. Спасибо редактору книги Виктории Живиной:
за моральную поддержку и важные советы.

CONCERT: ДЛЯ КОГО ЭТА КНИГА?

И ДЛЯ ЧЕГО?

Вступление

Эта книга ответит на самый главный вопрос: **как стать программистом Питона?** Я расскажу про язык Python, про его библиотеки, и про софт, которым нужно пользоваться. Кроме того, я буду рассказывать про людей в компьютерной сфере. Каково это — быть программистом: чем занимаются, чем живут и что умеют. Более того, в книге-учебнике будут лайфхаки о том, как сохранить мотивацию, распределить силу и расставить приоритеты.

Язык Python признан **самым популярным** и эффективным языком программирования в 2021 году. Он сумел обойти Java и C — очень серьезных соперников, годами не пускавших никого на первое место во всевозможных рейтингах. В чем его преимущество: огромный спрос на программистов в целом, он очень распространен и, наконец, легок в изучении.

Это неприятно, но, чтобы стать программистом — надо иметь **мозоль на заднице**. Высиживать перед компьютером по 12 часов буквально на чистом энтузиазме. Про то,

что из-за постоянной смены материалов и нововведений, учиться и обновлять знания придется всю жизнь. Каждый год появляется что-то новое, а старое безвозвратно уходит в историю. Иными словами, программисты постоянно изучают новое. Если тебе не нравится сидеть перед компьютером — ты не сможешь долго продержаться в таком режиме.

Данная книга написана еще и для того, чтобы ты смог узнать побольше о самой профессии программиста. **Решить для себя**, нужно тебе такое ремесло или нет. Я помогу тебе выбрать для себя то, что действительно тебе подойдет; то, где твои таланты смогут по-настоящему раскрыться. Способ познания лишь один — пробовать как можно больше разных вариантов, чтобы найти свой путь. Путь, выбранный сердцем (и немного разумом, естественно).

Даже если ты решишь выбрать дорогу, не связанную с IT, ты не уйдешь отсюда с пустыми руками. Есть множество полезных навыков, которым можно научиться у программистов и успешно применять их в других областях. Эта книга о сильных сторонах программирования. Например, как превращать **сложные задачи в простые**. Если ты сможешь осознать их, то получишь преимущество в любой области, которую выберешь. Познай секреты программистов, пока другие тебя не опередили.

Как устроена эта книга

Книгу можно **читать по диагонали**. Ее даже стоит прочесть несколько раз подряд, чтобы каждый раз находить что-то новое. У каждой главы есть определенная глубина. Можно читать только первый раздел, а затем переходить

к следующей главе, чтобы прочитать книжку по верхам и составить общее впечатление.

В том случае, если ты решил **вдумчиво изучить** одну главу — ты можешь задержаться и проникнуть в тему действительно глубоко. Где-то на последних разделах у каждой главы есть тексты, которые я писал только для профессионалов. Есть раздел, который требует усилий от читателя. То есть не просто почитать теорию, воспарив над схваткой, а с головой погрузиться в практику (получив боевой опыт в программировании). С руками по локоть в коде.

P.S. Если по какой-то причине ты все еще не знаешь английского, начни **его изучать** прямо сегодня! Без английского ты точно не сможешь разобраться в программировании. Еще никому не удавалось стать программистом без знания английского.

P.P.S. Без Linux тоже не обойдется. С **пингином** надо дружить!

Зачем нужна каждая из глав

В предпоследнем разделе каждой главы я буду давать задания, выполняя которые ты постепенно разберешься с Питоном. Даже если они покажутся тебе легкими — не пренебрегай ими. Сложность заданий **растет как снежный ком!**

Я расскажу те вещи, которые понадобятся тебе в работе. Лучше ты узнаешь ее от меня, чем будешь собирать по крохам в интернете. Без этой информации у тебя не будет пол-

ной картины. А ведь именно этим профи и отличается от новичка — он видит **картину в целом**. Мы вместе рисуем карту профессии, чтобы ты смог сам решить, чего тебе еще не хватает и куда двигаться дальше.

Вторая важная задача — мотивация и поддержка. Расскажу лайфхаки, которые помогут тебе **добраться до цели** и продолжить путь дальше; поделюсь вещами, которые помогают мне проходить сквозь стену непонимания нового предмета.

Каждая глава — это отдельное направление в ремесле программиста. Я хочу показать тебе перспективу и помочь сделать **первые шаги**. Моя книга — это именно карта профессии, а не инструкция. Опишу течения и рифы, но плыть тебе придется самому.

Как выжать максимум

Прочти книгу **несколько раз**. Я серьезно. С каждым прочтением тебе будут открываться все новые и новые смыслы, ведь тема слишком сложная, чтобы осилить ее с первого раза.

Этот пирог придется понадкусывать с разных сторон, прежде чем ты доберешься до середины.

Подпишись на группы в ВКонтакте и Телеграме, чтобы стать частью сообщества посвященного этой книге. Там публикуются отзывы читателей, я отвечаю на вопросы, а все вместе мы **хвастаемся своими достижениями**. Дружеская атмосфера. Вливайся!



Эффективность обучения вырастет в 3 раза, если ты будешь **выполнять задания** из книги. Более того, ты сможешь получить достижения, которые перечислены в конце книги. Чем больше достижений (ачивок, если говорить нашим языком) ты соберешь — тем больше удовольствия получишь от путешествия с книгой по волшебному миру Питона.

Чтение дополнительной литературы — **последний бастион**. Я не ожидаю, что из читателей вообще кто-то до него доберется. Когда прочитаешь книги из списков — напиши мне. Автор порадует за тебя и воздаст тебе почести. Адрес группы я указал вышел — добавляйся!;)

Другие источники

Сверхцель книги — это научить тебя пользоваться разными источниками.

Django Girls — если тебе нужен легкий вход для программирования на Питоне, то это оно. Тот самый сайт, с которого автор начинал свое изучение Python.

Спецификация Питона и его библиотек — иногда полезно заглянуть в документацию и узнать еще парочку способов использовать язык. Увидеть карту языка в целом.

Курсы питона в открытом доступе. Если тебе нужна лекция по самым азам языка — их тысячи! Тебе просто надо найти подходящего автора, который объясняет основы языка именно в твоём стиле. Попробуй почитать разных авторов, каждый рассказывает немного иначе. Это придаст объёмность твоим знаниям, согласишься на предмет под разными углами.

StackOverflow — в конце концов, все вопросы по языку приведут тебя сюда. Python — слишком популярный язык, поэтому на сайте скопилось фантастическое количество ответов на любые вопросы по Питону.

Yandex — в любой непонятной ситуации, спрашивай у Яндекса. Нынешнее поколение поисковых роботов умеет понимать даже самые смутные тени мыслей и давать на удивление хорошие ссылки. Если не получилось с первого раза, то попробуй переформулировать и спросить по-другому.

Лайфхак: для обзорных вещей помогает поиск по картинкам. Многие штуки можно зацепить, если последовать за удачной картинкой из какой-нибудь презентации.

Из полезного также:

1. Бесплатный курс от Яндекса;
2. основы Python от Яндекса;
3. платный курс от Яндекса.

Т.е. сначала можно пройти бесплатный курс, если готов продолжать — уже подходить к платному обучению. По окончании платного, скорее всего, найдёшь работу.

ChatGPT — новомодная штука. Кажется, что следующие поколения программистов будут активно пользоваться ею, чтобы получать резюме по любому объёмному чтиву.

Удобная штукавина для поверхностного знакомства. Обрати внимание на отечественную разработку: GigaChat.

Самый удобный интерфейс — это чат-боты в Телеграм. Есть бесплатные, есть за деньги, но обычно хватает бесплатных. Сам робот ничего не придумывает, он просто компилирует выжимку из доступных ему источников, придумывая недостающее. Будь осторожен! Робот **местами привирает**, да так ловко, что не сразу разберешься. За ним надо все тщательно перепроверять. Прочитал код робота? Напиши свой, который будет еще лучше!

Контрольные вопросы:

1. Чем занимаются программисты?
2. Чему можно научиться у программистов?
3. Хочешь ли ты стать программистом на Питоне?
4. Как лучше всего читать эту книгу?
5. Где спросить, если что-то не понятно?

Если ответил на все вопросы, получаешь достижение **#Concept**.

В сухом остатке:

1. прочитай книгу несколько раз;
2. выполняй задания в конце каждой главы;
3. работай над мотивацией, чтобы достичь своей цели;
4. изучай английский и Linux;
5. пользуйся разными источниками.

AGILE: КАК ПЕРЕСТАТЬ ЧИТАТЬ И НАЧАТЬ ПРОГРАММИРОВАТЬ. КАК ПОБОРОТЬ ЛЕНЬ. ДОФАМИН.

Что может пойти не так

Какие сложности могут быть **при восхождении** на самую обычную гору? Или при прочтении учебника? Обычное дело, за которое со временем берется все меньше и меньше людей. Там же вроде все просто, не так ли?

Я не говорю, что чтение учебника — это подвиг. Но и легким времяпрепровождением я это не назову. Это работа, требующая усилий. **Учись уважать** свой (и чужой) труд, нельзя обесценивать свои усилия. Если ты не признаешь, что тратишь силы, то и восстанавливать их не будешь. Откуда тогда брать энергию для свершений?

Бери и делай! Примерно у 95% процентов на этом этапе начинаются проблемы. Проблемы, о которых как-то **неловко говорить**, ведь окружающие поднимут на смех. Ты и сам наверняка посмеивался над этими неудачниками. Бывает очень тяжело признаться, что столкнулся с трудностями. Нежелание делать — еще один способ избежать работы.

Делать — скучно. Ты тратишь усилия, а мгновенного результата что-то не видать. Сначала ты просто запоминаешь кучу бесполезных вещей, прежде чем их наберется достаточно для того, чтобы между ними возникли связи и появилась цельная картина — знание и навык. Вопрос в том, как много бесполезных вещей ты способен удержать, прежде чем мозг нажмет кнопку стирания памяти, и переключится на что-то другое.

Если **постоянно не получается** — это вообще удар ниже пояса. Зачем делать то, что не получается? Суть в том, что как только ты начинаешь делать что-то новое — случаются ошибки. Иначе ты бы давно это делал. Есть отличный способ не делать ошибки — просто не делать ничего нового. Никогда не пытайся выйти за границы своих возможностей и все будет в порядке (сарказм). Как люди становятся успешными? Они совершают ошибки, намного больше ошибок, чем обычные люди. Секрет в том, что они не останавливаются. Продолжают пытаться и совершать все новые и новые попытки. Однажды у них это получается. Или нет. Я считаю, что стоит рискнуть!

Потому что **мотивация на нуле**, а у всех людей разный уровень дофамина. И если чемпион сможет вывезти на характере, то обычный человек все это бросит при первом же затруднении. Пойдет смотреть мультики. Над депрессией смеются только те, кто никогда ее не испытывал: когда мир теряет краски; еда на вкус как картон; а внутри пустота, которую ничем не заполнить, ведь все бессмысленно. Это вопрос химии в голове, такое нельзя просто отменить. Нельзя просто встать и пойти, **если тебе прострелили колено**, но люди вообще-то очень упорные. Есть лекарства, есть программы реабилитации, которые помогают вылезти из этой ментальной западни.

У всех бывают минуты слабости, вопрос лишь в том, хочешь ты из этого выбраться или хочешь остаться там, где ты сейчас.

Есть мечта — беги к ней.

Не можешь бежать — иди к ней.

Не можешь идти — ползи к ней.

Не можешь ползти — ляг и лежи в сторону мечты.¹

Боль и унижение — часть настоящей учебы. Боль — потому что у тебя ничего не получается. Унижение — потому что ты такой тупой. Наверняка окружающие смеются над твоими жалкими попытками! На самом деле нет. Каждый раз, когда ты будешь испытывать неприятные эмоции, знай, что ты не первый, и не последний, кто проходит этот путь. Прошли и оставили позади, это не будет длиться вечно. Любому росту предшествует что-то похожее. Отчаяние и разочарование — **признаки начала роста**, так устроен наш мозг. Чтобы поменяться, надо осознать невозможность оставаться прежним.

Мы рубим ступени. Ни шагу назад!

И от напряженья колени дрожат,

И сердце готово к вершине бежать из груди.

Весь мир на ладони — ты счастлив и нем

И только немного завидуешь тем,

Другим, у которых вершина ещё впереди.²

В этой главе расскажу пару лайфхаков, как преодолеть долину боли и унижения; откуда брать **мотивацию**

¹ Полозкова Вера. «Стихи из онлайн», 2013–2017.

² Владимир Высоцкий. Из фильма «Вертикаль». 1966.

и силы для следующего шага; как помочь себе в минуты слабости; куда вкладывать силы в минуты энтузиазма. Чтобы вынырнуть на той стороне и, наконец, запустить свою первую программу на Python. Это сладкое чувство, когда твой код наконец скомпилировался без ошибок и запустился...

Первый шаг

Самый первый лайфхак: делай **Minimal Value Product** (MVP). Что даст результат, но при этом не потребует много усилий? Первый результат поможет тебе понять, что ты на верном пути. Когда есть успешный опыт, его легче повторить и развивать дальше.

В нашем случае это программа из одной строчки:

```
print ('Hello World')
```

Для начала будет достаточно написать эту строчку в браузере на сайте с он-лайн компилятором Питона (их тысячи!). Я просто поискал в Гугл «**Python online**» и открыл первый попавшийся.

Кстати, там уже была написана эта строчка для примера работающей программы — очень удобно!

Если у тебя получилось написать первую программу, получай достижение **#HelloWorld**.