



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

برنامه سازی پیشرفته و کارگاه

شیوه نامه داکیومنت نویسی

استاد درس

دکتر مهدی قطعی

استاد دوم

بهنام یوسفی مهر

نگارش

سیدآرمان حسینی

بهار ۱۴۰۳

فهرست

3	مقدمه
4	محتوا
4	Interactive بودن داک‌ها
4	سریع به کد برسین
5	«چه چیزی یاد گرفتیم؟»
6	استفاده از کلمات انگلیسی
7	لحن
7	فارسی روان و خودمونی
7	استفاده از ChatGPT
9	ظاهر
9	برای headerها از word استفاده کنین
10	به template وفادار باشین
10	کدها و کامندها
11	فاصلهٔ بین پاراگراف‌ها
12	متن رو justify کنید
12	فهرست
13	خیلی زیر عکس‌ها چیزی ننویسید
13	از bullet pointها استفادهٔ بی‌جا نکنید
14	همهٔ متن راست‌چین شده باشه
14	اسم‌گذاری
15	چک و تیک‌های نهایی

مقدمه

این شیوه‌نامه، برای تعیین یک لحن و شیوهٔ ثابت و هماهنگ بین داکيومنت‌های مختلف تدریس‌یاری نوشته شده. توی این شیوه‌نامه، قواعد و ساختارهای مختلفی که بین همگی داکيومنت‌ها رعایت می‌شه رو بررسی می‌کنیم.

با توجه به اهمیت داکيومنت‌ها توی درس AP، لازمه که همهٔ تدریس‌یارها اون رو بخونن و توی نوشتن داکيومنت‌هاشون، ازش استفاده کنن. در فرآیند داک رییوها، حتما بخش‌های مختلف این شیوه‌نامه بررسی می‌شن.

محتوا

Interactive بودن داکها

داکيومنت‌ها باید تا حد ممکن interactive باشند. از بچه‌ها بخواین که ادیتور یا ترمینال‌شون رو باز کنن و کدهای توی داکيومنت رو توشون بززن. اگر دارین مبحثی مثل دیباگر رو درس می‌دین، از بچه‌ها بخواین که بریک‌پوینت بذارن، دکمه دیباگر رو بززن و توقف برنامه رو ببینن.

اصلی‌ترین نکته‌ای که توی داکيومنت‌ها باید رعایت بشه، همین interactive بودنشونه. داکيومنت‌ها، lecture نیستن. اگر می‌بینید که چند صفحه‌ست که دارین فقط با بچه‌ها صحبت می‌کنین و هیچ کدی نزدین، شاید یک جای کار می‌لنگه.

سریع به کد برسین

توی داکيومنت‌ها، تا جای ممکن سعی کنید که سریع به کد و کارهای عملی برسین. مثلاً فرض کنید که می‌خوایم یه داک راجع به دیباگر بنویسیم. می‌تونیم این داک رو به دو شکل بنویسیم:

1. یکی دو صفحه از اهمیت دیباگر بنویسیم، بعدش توی دو صفحه بعدی از انواع روش‌های دیباگ کردن بگیم، یک صفحه هم راجع به بریک‌پوینت و انواعش صحبت کنیم، نهایتاً بعد از پنج یا شش صفحه به بچه‌ها بگیم که واقعا برن بریک‌پوینت بذارن.
 2. همون صفحه اول یا دوم، بدون هیچ مقدمه خاصی، به بچه‌ها بگیم که روی اون دایره قرمز کنار خط‌ها (که هنوز نمی‌دونن بریک‌پوینته) کلیک کنن. بعدش روی اون دکمه سوسک‌مانند بالای صفحه (که نمی‌دونن دکمه دیباگه) کلیک کنن. بچه‌ها می‌بینن که برنامه‌شون شروع می‌شه و یهو توی یه نقطه متوقف می‌شه! مقادیر متغیرها کنارشون نوشته شده، اون پایین چندتا دکمه جدید جای خروجی اضافه شده و ...
- حالا که بچه‌ها دیدن که برنامه متوقف شده و براشون سوال شد که چه اتفاقی افتاده، بهشون بریک‌پوینت و دیباگ کردن رو توضیح می‌دیم.

توی روش دوم، شما خیلی خوب تونستین توجه بچه‌ها رو جلب کنین و اون‌ها رو با مطلب همراه کنین، در حالی که توی روش اول، بیشتر حوصله‌شون رو سر بردین.

ما از شما، روش دوم رو می‌خوایم. نترسید از این که از نقطه‌ای که بچه‌ها هنوز کامل بلدش نیستن، داک رو شروع کنین. ممکنه شما برای شروع داکيومنتی که قراره گرافیک رو یاد بده، از بچه‌ها بخواین که بدون دونستن این که JFrame و JLabel چیه فقط کد زیر رو کپی کنن و خروجی‌ش رو ببینن:

```
import javax.swing.*;

public class Main {
    public static void main(String[] args) {
        var frame = new JFrame("Hello, java");
        frame.setSize(300, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        var label = new JLabel("Hello, Java!", JLabel.CENTER);
        frame.add(label);

        frame.setVisible(true);
    }
}
```

بعد از دیدن خروجی، تازه بهشون توضیح می‌دین که frame چیه، label چیه و خلاصه این که اصلا این کد چه کار می‌کنه. اتفاقاً این روش شاید خیلی بهتر برای بچه‌ها جواب بده، به جای این که داکتون رو از توضیح روی هوای این که JFrame و JLabel چیه شروع کنین.

حواستون باشه که داکيومنت‌ها، یک ارائهٔ خشک و خالی یا یک متن نامربوط نیستن. داکيومنت‌ها لیتراالی معلم بچه‌هان، و یه جورایی هم، تنها معلم بچه‌هان. همیشه تصور کنید که اگر بنا بود کنار یکی‌شون بشینید و راجع به موضوع داک بهش یاد بدین، چه کار می‌کردین، همون رو مکتوب کنین. مثلاً احتمالاً ایدهٔ بدیهه اگر نیم ساعت اول کلاستون رو راجع به تاریخچهٔ فلان library صحبت کنید، پس این‌جا هم ایدهٔ بدیهه.

«چه چیزی یاد گرفتیم؟»

در انتهای هر داکيومنت، توی یه بخش جدا به اسم «چه چیزی یاد گرفتیم» محتوایی که بعد از خوندن این داک لازمه بچه‌ها بلد باشن رو مرور کنید تا اگر به خوبی یادش نگرفتن، از تدریس‌یارها سوال بپرس و دوباره داک رو بررسی کنن.

استفاده از کلمات انگلیسی

توی داکيومنت‌ها، حتما از معادل انگلیسی کلمات استفاده کنین. مثلا اگر دارین abstract class رو توضیح می‌دین، از عبارتی مثل «کلاس انتزاعی» استفاده نکنین. هم سرچ بچه‌ها رو سخت‌تر می‌کنین و هم آشنایی‌شون با زبان انگلیسی رو به تاخیر می‌ندازین. اگر حس می‌کنید که شاید بعضی‌ها ندونن معنی اون کلمه چیه، بار اولی که توی متن استفاده می‌کنیدش به شکل «کلاس انتزاعی (abstract class)» به کارش ببرین و بعد از اون دیگه فقط abstract class رو بنویسید.

کامنت‌ها، راه خوبی‌ان برای این که کدهاتون رو توضیح بدین، ولی اگر کامنتی می‌ذارین، حتما انگلیسی باشه.

لحن

فارسی روان و خودمونی

داکیومنت‌ها، باید خودمونی و به فارسی روان نوشته بشن. همچنین، لحن اون‌ها باید دوم شخص جمع باشه. جملاتی مثل جمله زیر، برای داکيومنت خوب نیستن:

- **«استفاده مؤثر از ابزارهای IDE می‌تواند سرعت و دقت شما را در حل مسائل به شکل چشمگیری افزایش دهد.»:** این جمله، زیاده از اون چیزی که می‌خوایم رسمیه و فارسی نوشتاریه بیشتر تا گفتاری.
- **«استفاده مؤثر از ابزارهای IDE می‌تونه سرعت و دقت رو توی حل مسائل خیلی خیلی زیاد کنه.»:** این جمله، خیلی دیگه خودمونی! دوم شخص جمع نیست و می‌تونه بهتر نوشته بشه.
- **«اگه از ابزارهای IDE درست استفاده کنین، سرعت و دقتتون توی حل مسائل خیلی بیشتر می‌شه.»:** این جمله، خیلی جمله بهتریه. هم روان‌تره، هم به اندازه کافی خودمونی، و دوم شخص جمعه.

همچنین، حتما حواستون باشه که از نیم‌فاصله‌ها و علائم نگارشی به درستی استفاده کنین. توی این مورد، چت جی‌پی‌تی بهتون کمک می‌کنه.

استفاده از ChatGPT

لطفا همیشه، بعد از اتمام داکيومنت، جملات اون رو با ChatGPT خواناتر کنید. عموماً خروجی ChatGPT خیلی خواناتر از چیزیه که شما نوشتید که دنبال کردنش برای بچه‌ها راحت‌تره. من خودم پاراگراف به پاراگراف متن رو با prompt زیر به ChatGPT می‌دم تا متن بهتری بهم بده. گاهی لازمه وسط راه به مقدار راهنمایی‌ش هم بکنید:

پاراگراف‌هایی که در ادامه بهت می‌دم، مربوط به یه داکيومنت کلاس برنامه‌نویسیه.
می‌خوام اون‌ها رو به فارسی روون و خودمونی، ولی نه خیلی خودمونی دوباره بنویسی و برام بفرستی

البته! پاراگراف‌ها رو بفرست، من اون‌ها رو به فارسی روان و طبیعی بازنویسی می‌کنم.

در برنامه‌نویسی، استفاده مؤثر از ابزارهای IDE می‌تواند سرعت و دقت شما را در حل مسائل به شکل چشمگیری افزایش دهد. در IntelliJ IDEA، مجموعه‌ای از امکانات قدرتمند برای بهبود تجربه کدنویسی و رفع باگ‌ها و اشکالات برنامه فراهم شده است.

در برنامه‌نویسی، اگر از ابزارهای یک IDE به درستی استفاده کنید، می‌توانید سرعت و دقت خود را در حل مسائل تا حد زیادی بالا ببرید. IntelliJ IDEA امکانات پیشرفته‌ای دارد که تجربه کدنویسی را بهتر می‌کند و به شما کمک می‌کند باگ‌ها و مشکلات برنامه را راحت‌تر برطرف کنید.

یکم خودمونی‌تر، دوم شخص جمع بنویس. فارسی رسمی نباشه.

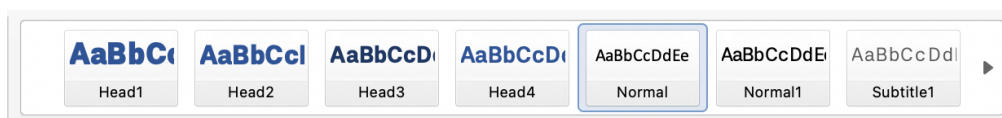
اگه از ابزارهای یه IDE درست استفاده کنین، سرعت و دقتتون توی حل مسائل خیلی بیشتر می‌شه. IntelliJ IDEA کلی قابلیت کاربردی داره که هم کدنویسی رو براتون راحت‌تر می‌کنه، هم کمک می‌کنه باگ‌ها و مشکلات برنامه رو سریع‌تر پیدا و برطرف کنین.

با توجه به این که خیلی وقت‌ها، چند نفر یک داک رو می‌نویسن، حواستون باشه که لحن داک بین بخش‌های دو نفر مختلف تغییر نکنه. ChatGPT توی این مورد هم به شما کمک می‌کنه و لحن یه داکيومنت رو ثابت می‌کنه.

ظاهر

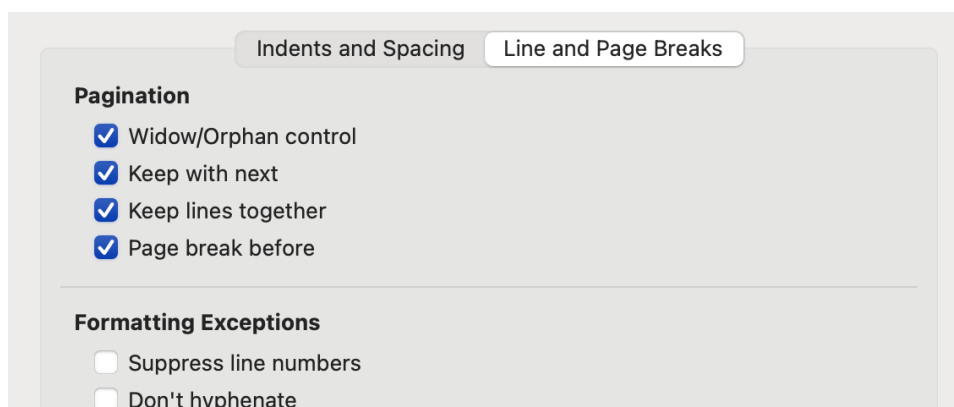
برای headerها از word استفاده کنیم

اگر می‌خواهید راحت فهرست درست کنیم و هدرها را خوب توی اون بیوفتن، برای تعیین هدر هر بخش از خود word کمک بگیرید:



شما باید Head1 را برای عناوین خیلی مهم، Head2 را برای زیرعنوان‌ها، و Head3 را برای چیزهای کوچک‌تر نگه دارید. مثلاً اگر دارید متدهای مختلف کلاس‌های گرافیکی جاوا رو توضیح بدین و جایی توی اون به JFrame رسیدین و می‌خواهین متد `setSize(int x, int y)` رو توضیح می‌دین، به ترتیب برای «متدهای کلاس‌های گرافیکی» از Head1، برای «JFrame» از Head2 و برای «`setSize(int x, int y)`» از Head3 استفاده کنیم. حواستون باشه که متن نرمال داک از Headها استفاده نکنه چون برای درست کردن فهرست به مشکل می‌خورین.

برای هر Head1، شما باید یه صفحهٔ جدید رو شروع کنیم. مثلاً توی همین شیوه‌نامه که جلوتونه، هر بخش «مقدمه»، «ظاهر»، «لحن» و «محتوا» یه صفحهٔ جدید رو شروع کرده. برای این که بتونید این کار رو بکنید، روی اون هدر کلیک راست کنید، Paragraph... رو بزنید و توی صفحه‌ای که باز می‌شه، از بخش Line and Page Breaks تیک Page break before رو بزنید:

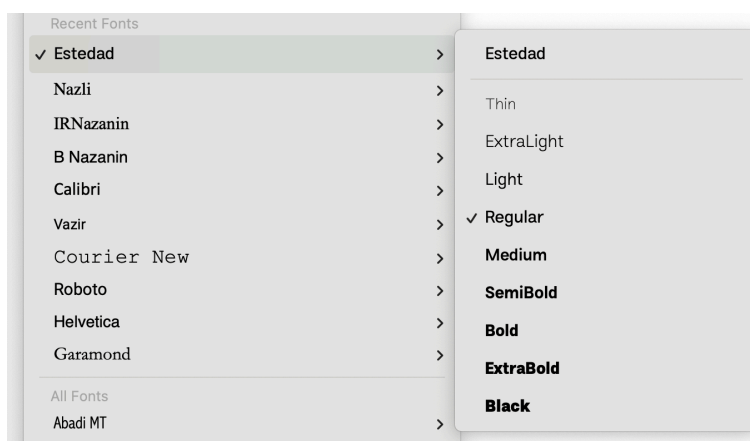


به template وفادار باشین

توی داکيومنت‌ها، به هیچ وجه template و اجزای اون رو عوض نکنین. چینش اجزای صفحهٔ اول رو جابه‌جا نکنید و از فونت دیگه‌ای جز استعداد و Courier New استفاده نکنید.

ترجیحا تکست‌ها رو با رنگ یا شکل متفاوتی نسبت به باقی داک‌ها ننویسید، مثلا اگر شما توی داکتون کلمات مهم رو قرمز کنید، احتمالا کس دیگه‌ای توی داکش این کار رو نکنه و هماهنگی بین داکيومنت‌ها از بین بره.

دقت کنید که ما متن تمام داکيومنت‌ها رو با فونت استعداد، وزن Regular و سایز ۱۲ می‌نویسیم. اگر جایی نیاز دارین متنی رو بولد کنین، از دکمهٔ بولد خود word استفاده نکنید و به جاش، توی خود وزن‌های مختلف استعداد، وزن Bold رو انتخاب کنید:



تفاوت بولد خود ورد و بولد استعداد توی این‌جا مشخصه، و بولد استعداد بهتره. حواستون باشه که ازش استفاده کنین.

کدها و کامندها

برای این که کدهاتون، Syntax Highlighting داشته باشن، همیشه اون‌ها رو مستقیما از IntelliJ کپی کنید:

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

با این کار، هم فونت کدها یک‌دسته و هم کدتون مرتب پیست می‌شه. اگر جایی می‌خواین به command ترمینال رو توی داک بذارین، لطفا اون‌ها رو قبلش توی یه فایل sh. که توی IntelliJ درست کردین بنویسین و از اون‌جا کپی کنین، تا اون هم به خوبی نمایش داده بشه. همیشه قبل از دستورات command-line، به \$ بذارین:

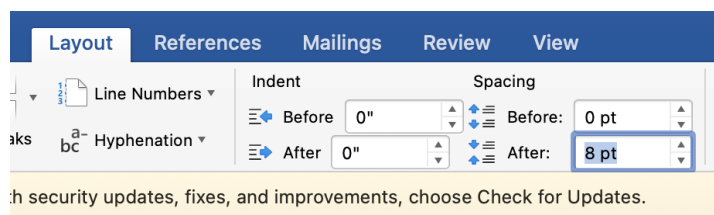
```
$ git version
```

اگر نیاز داشتن یک متن رو با این استایل توی وردتون بیارین، کافیه اون رو توی یه فایل txt. توی IntelliJ بنویسید تا با کپی شدنش، استایلش شبیه باقی کدها بشه:

```
commit 56f1a0e2c57a6451a0d465c6c502ad711c193d93 (HEAD -> main)
Author: Arman Hosseini <armanhosseini878787@gmail.com>
Date: Thu Feb 13 14:26:09 2025 +0330

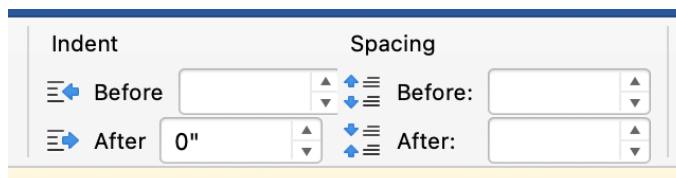
    update README.md
```

اگر دقت کنید اما، عموماً بعد از این کار، فاصلهٔ این کد تا خط بعدی‌ش خیلی کمه و داک تو هم تو هم می‌شه. لطفاً اگر این‌طور شد، کدتون رو انتخاب کنید، به بخش Layout برین و توی Spacing، بخش After رو روی 8 pt بذارین:



فاصلهٔ بین پاراگراف‌ها

همیشه بعد از نوشتن داکيومنت، چک کنید که فاصلهٔ بین پاراگراف‌ها همگی 8 pt باشه. شاید لازم باشه دو صفحه داک رو سلکت کنید و فاصله‌ها رو روی 8 pt تنظیم کنید. اگر کل داکتون رو سلکت کردین، و توی Spacing عددی ندیدین، یعنی یه جا هست که فاصله 8 pt نیست:



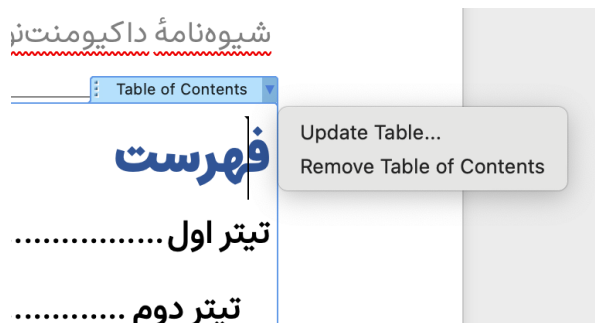
هیچ وقت اگر فاصلهٔ دو پاراگراف کمتر از چیزی که می‌خواستین بود، از اینتر اضافه استفاده نکنین و از همین روشی که توضیح دادیم تنظیمش کنین. چون که اینتر اضافه هم زشته، هم اگر یه بخش داک رو تغییر بدین و کوتاه یا بلندتر کنین ممکنه به هم بریزه.

متن رو justify کنید

حتما بعد از اتمام نوشتن داک، متن اون رو justify کنید. برای این کار از خود justify عادی استفاده کنید و justify low و سایر گزینه‌ها رو نزنید. حواستون باشه که کدها و عکس‌ها رو justify نکنید.

فهرست

برای فهرست، از فونت استعداد و سایز ۱۰ استفاده کنین. اگر هدرها و متن‌هاتون رو با ورد به درستی مشخص کرده باشین، با زدن دکمهٔ Update Table... باید فهرست به شکل خودکار براتون درست بشه:



لازمه که برای هدرهایی که از جنس Head1 ان از وزن Bold و برای باقی فهرست از وزن Regular استفاده کنید:

فهرست

3.....	مقدمه
4.....	محتوا
6.....	لحن
8.....	ظاهر
8.....	برای header از word استفاده کنین
9.....	به template وفادار باشین
9.....	کدها و کامندها
10.....	فاصلهٔ بین پاراگرافها
11.....	متن رو justify کنید
11.....	فهرست
11.....	خیلی زیر عکس‌ها چیزی ننویسید

خیلی زیر عکس‌ها چیزی ننویسید

عموما بهتره که زیر عکس‌های داک متن بلندی ننویسین و اگر می‌خواین عکسی بذارین، قبلش راجع بهش توضیح بدین. متن‌های زیر عکس داک، اگر وجود دارن خوبه که کوتاه و کوچیک باشن.

عموما عکس‌ها رو با یه دو نقطه، زیر چیزی که دارین راجع بهش توضیح می‌دین بیارین:



از bullet point ها استفاده بی‌جا نکنید

bullet point ها همیشه باعث یه tab نامربوط توی داک‌تون می‌شن که زشت‌ترش می‌کنه. مثال زیر رو در نظر بگیرین:

برخی از متدهای پرکاربرد برای کلاس‌های کاربردی Swing

متدهای کلاس JFrame

- `setText(String title)`

این متد، همون تایتل فریم رو تغییر میده

- `setSize(int width, int height)`

ابعاد فریم رو مشخص میکنه

- `setVisible(boolean state)`

وضعیت نمایش فریم رو مشخص میکنه

- `setResizable(boolean state)`

این‌جا، استفاده از بولت‌پوینت‌ها باعث شده که همهٔ متن یک تب به جلو بره و نظم داک رو به هم ریخته. توی این‌جا، استفاده از Head3 خیلی کار قشنگ‌تریه:

برخی از متدهای پرکاربرد برای کلاس‌های کاربردی Swing

کلاس JFrame

setText(String title)

این متد، همون تایتل فریم رو تغییر میده.

setSize(int width, int height)

ابعاد فریم رو مشخص می‌کنه.

setVisible(boolean state)

وضعیت نمایش فریم رو مشخص می‌کنه.

setResizable(boolean state)

اگر ورودی را false قرار دهیم، ابعاد فریم ثابت باقی‌مانده و کاربر نمیتواند آن را تغییر دهد.

همهٔ متن راست‌چین شده باشه

مثل مثال مورد قبلی، همهٔ متن باید راست‌چین شده باشه، حتی اگر فقط شامل یک عبارت انگلیسیه.

اسم‌گذاری

ما برای اسم‌گذاری داک‌هامون از kebab-case استفاده می‌کنیم. مثلاً اسم این داک که جلوتونه how-to-write-a-good-doc، نه HowToWriteAGoodDoc یا How_to_write_a_good_doc. لازمه شما هم از این convention پیروی کنید.

چک و تیک‌های نهایی

لازمه قبل از انتشار هر داکيومنت، موارد زیر رو چک کنید:

- چک کنید که تایتل داکيومنت‌تون در صفحهٔ اول، به درستی وارد شده باشه.
 - اسم نویسندگه‌ها همگی باید بالای داکيومنت‌تون باشه. حتما اسم سوپروایزر اون هفته هم توی داکتون بیارین که بچه‌ها بدونن راجع به محتوای اون داک می‌تونن ازش سوال بپرسن.
 - چک کنید که همهٔ صفحات داک، شماره صفحه داشته باشن. خیلی وقت‌ها از یک صفحه‌ای به بعد ورد براتون شماره صفحه نمی‌دازه.
- برای درست کردن این مشکل، کافیه بعد از اتمام داکيومنتتون، از بخش Footer گزینهٔ Banded رو انتخاب کنید:

