大丈夫。

Dartマップとその機能、作成方法、使用目的について簡単に説明しましょう。 Dartマップについて話しているとき、それは一種の世界のマップではなく、何かが他の何かにマップできる方法のようなものを指します。

たとえば、この6月の経度をGoogleマップに入力すると、特定の場所にマッピングされます。

それよりもさらにいいのは、地図を電話帳のようなものだと考えるべきでしょうか？

私には3人の友人がいるとしましょう。

私よりも多くの友人がいることを願っています。

エイミー、ジェームズ、そしてティムと私は、そのような古い学校だから、電話番号を本に残しています。

今、電話帳は非常に基本的な種類の地図のようなものです。

2つのデータを関連付けることができます。

したがって、通常はキーがあります。この場合、それは私の友人の名前になり、その後、いくつかの値があり、これらはキーに関連付けられた値になります。

これがマップの仕組みです。これらは、以前見たリストのようなコレクション型のようなものです。

ただし、リストとは異なり、マップには順序がありません。スーパーOCDで、アルファベット順に並べて、毎回電話帳を書き直さない限り、電話帳で名前を並べるのは意味がないので、0から1から2にはなりません。 Aで始まる名前の友人がいました。しかし、本質的にすべてのマップはキーと値があり、キーは値にマップされ、キーがわかっていれば値を引き出すことができます。コードでは、これは次のようになります。

したがって、それらは中括弧のセット内に含まれます。

キーがあり、コロンの後にそのキーに関連付けられた値があります。そして、これらのエントリはすべてコンマで区切られています。

これが作成方法であり、使用方法です。マップを作成する場合は、データ型をマップデータ型として指定できます。その後、一連の山かっこを使用できます。データ型を入力するリストがあるときと同じように、リストでは、この場合、キーのデータ型と値のデータ型を指定して、マップを作成するとき、またはマップに値を追加しようとするときにコンパイラがキーを確認できるようにチェックできます最初のキータイプに適合し、値が値タイプに適合します。

マップを使用する場合は、マップの名前を参照し、角かっこを使用してキーを提供します。そして、これにより、そのキーに関連付けられている値が提供されます。

それでは、新しいマップをゼロから作成しましょう。

まず、マップデータタイプになるデータタイプを指定することから始めます。そして、そのままにして、マップ内のエントリのデータ型が何であるかをDartに把握させます。

しかし、もし私がそれを助け、またこの特定のマップの中に含まれているものを一目で見ることができるように私が一緒に働いている他のプログラマーを助けたいなら、いくつかの山括弧を追加することができます。内部では、最初にキーのデータ型を提供できます。この場合、キーは友人の名前である文字列になります。そして、値は整数、つまりそれらの人々の電話番号になります。

そして、これはその完全なデータ型がどのように見えるかです。型マップのコレクションを作成していますが、これには文字列になるキーがあり、整数になる値が関連付けられています。そして、名前を付けます。

それでは、電話帳を作成しましたか？

そして、新しいマップと等しくなるように設定されます。

そこで、中かっこをいくつか開き、ここでキーと値を指定します。

この場合、キーは文字列でなければなりません。

だから、私は知らないキーと呼ばれるキーを作成し、これがキーの終わりであることを示すためにコロンを追加するとしましょう。

そして今、カイルに関連付けられる値を書きますので、カイルがランダムな電話番号にいるとしましょう。そして、コレクション内の各エントリをカンマで区切ることができます。

それでは、時間の都合上、前の基調講演からのものを貼り付けます。

そして今、セミコロンでマップの終わりを埋めれば、きれいに見えるようにマップを再フォーマットできます。

したがって、ここには文字列と整数のマップがあります。これは電話帳と呼ばれ、中括弧のセットに含まれており、4つのエントリがあります。

それで、私にできることは、地図を使い始めることができるということです。

たとえば、電話帳にある値を印刷したい場合は、エイミーが正しいとしましょうか？

次に、電話帳を作成し、コレクションから何かを引き出しようとしていることを示す角かっこを追加し、マップのキーの1つと明らかに一致する必要があるキーを提供します。

それでは、Amyの電話番号をコンソールに印刷してみましょう。実行してみましょう。123456789を取得することがわかります。これは、キーAmyに関連付けられている値です。

マップに実際に存在しないキーを取り出そうとした場合、この電話帳から自分の番号を取得しようとしたとすると、取得できるのはnullだけです。存在しないキーの値を取得しようとすると、値がnullになります。

したがって、特定のキーが存在するかどうかを確認したい場合は、これに対して簡単にチェックできます。特定のキーの電話帳がnullに等しいかどうかを確認できます。これを実行します。

それ以外の場合は。

また、この特定の構文（角かっこ）を使用して、まだ存在していない新しい値を電話帳に追加することもできます。

それで、電話帳を書いて、自分の名前に値を追加するとしましょう。

それで、アンジェラが等しいキーに電話帳を追加するつもりです。いくつかの乱数を追加しましょう。

そして、キーのアンジェラの電話帳をもう一度印刷しようとすると、これが機能するようになり、ここでnullが印刷されなくなります。したがって、マップの素晴らしい点は、順序付けられていないことです。

0がカイル、1がエイミーのように順番に取り出す必要がないので、エントリがどの時点で入るかは本当に重要ではありません。代わりに、一目で見たときに非常に読みやすく、理解しやすいものを使用して、実際にマップから値を取得できます。

この構文は非常に表現力豊かです。

現在、マップを簡単に使用できるようにする多くのメソッドがあります。

したがって、たとえば、phonebook.lengthを記述して、電話帳に現在あるエントリの数を確認できます。

つまり、4とこれに加えて5です。

または、phonebook.keysを記述して、電話帳にあるすべてのキーを印刷できます。そして、それは現在持っているすべてを印刷します。または、値に対しても同じことを行うことができ、それが持っているすべての値を出力します。ですから、今後のレッスンでもっと多くの地図を見ることになるでしょう。

また、特定の順序ではなく、キーで検索可能なアイテムをコレクションに入れる便利な方法です。