Formeln in DENTPORT

FORMELN VERSTEHEN, KORREKT NUTZEN UND SELBST ERSTELLEN

Einführung 2

Ein einfaches Beispiel 3

Sprache bzw. Syntax **6**

Funktionen 8

Variable 9

Formeln für Texte **9**

Noch ein Beispiel für eine Textformel 10

Formeln für Formulare **12**

Anwender- und Hintergrundformeln **16**

Ein Beispiel für eine Anwenderformel 16

Hintergrundformeln 19

Im- und Export von Formeln **19**

Einführung

DENTPORT bietet mit der Möglichkeit, Formeln zu erstellen, ein sehr leistungsfähiges Werkzeug, um die Fähigkeiten des Programms zu erweitern. Mit Formeln lassen sich eine ganze Reihe von Problemen lösen. Natürlich ist es normalerweise nicht nötig, sich mit dem Erstellen von Formeln auseinander zu setzen, zumal dies auch recht fehlerträchtig sein kann. Aber es kann nicht schaden, zu wissen, welche Aufgaben sich mit Formeln erledigen lassen – auch dann, wenn man womöglich nicht komplett versteht, wie das im Einzelnen funktioniert.

Es kommt durchaus vor, dass die dentport GmbH einzelne Wünsche nach Funktionen oder Features mit dem Versenden einer Formel beantwortet. Wir wollen diese Formeln auch anderen Nutzern der Software anbieten, so dass alle davon profitieren können. Dieses Tutorial soll Sie aber dabei unterstützen, diese Formeln auch zu verstehen und womöglich an Ihre eigenen Bedürfnisse anzupassen.

Formeln können mehrere Funktionen erfüllen: Die offensichtlichste Anwendung ist die Definition zusätzlicher Textvariablen. Doch es gibt noch weitere Formeln, die nochmals mächtiger sind: Für Formulare und Extra Daten lassen sich spezielle Formeln definieren, die Eingaben überprüfen und auswerten. Anwenderformeln fügen in bestimmten Bereichen neue Knöpfe hinzu und können Datenbanken abfragen, Eingaben überprüfen oder andere Berechnungen durchführen, was auch im Hintergrund ablaufen kann. Anwenderformeln sind eher ein Thema für fortgeschrittene Anwender oder aber für Mitarbeiter der dentport GmbH, denn sie erfordern meist ein tieferes Verständnis der Datenbankstruktur, in der DENTPORT Informationen speichert.

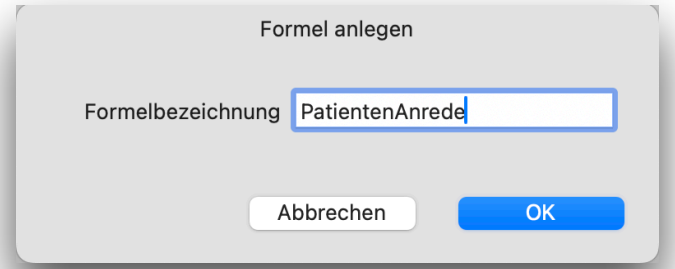
Keine Bange: Von richtiger Software-Entwicklung ist das Erstellen von Formeln ein gutes Stück weit entfernt und einfachere Aufgaben sind auch ohne vertiefte Kenntnisse zu bewältigen, zumal alle Aspekte und Elemente der Formelentwicklung direkt im Programm erläutert werden. Zumindest sollten Sie alle Formeln, die wir Ihnen liefern, auch verstehen können. Da Formeln auch Textvariable enthalten können, sollten Sie sich damit bereits auskennen. Der Gebrauch und Inhalt von Textvariablen wird in einem anderen Tutorial näher erläutert.

Ein einfaches Beispiel

Starten wir mit einem einfachen Beispiel, für dass niemand ernsthafte Programmierkenntnisse benötigt. Entstehen soll eine einfache Formel für die korrekte Anrede für jedes Geschlecht, die sich hinterher wie eine Textvariable nutzen lässt.

Um eine neue Textformel anzulegen, gehen Sie an eine Stelle, an der Sie Textvariable nutzen können (z.B. Stammdaten>Vorlagen>Textvorlagen>Patientenbrief), blenden Sie die Textvariablen ein und drücken Sie die Schaltfläche „Formeln …“.

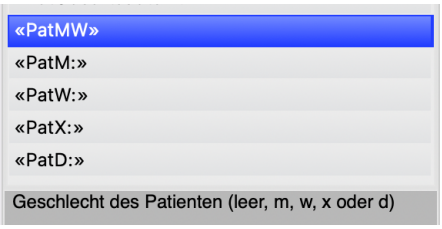
Es öffnet sich der Dialog zum Erstellen von Formeln, in dem Sie rechts die Formel selbst eingeben können. Auf der linken Seite finden sich vier Reiter (***Formeln, Syntax, Funktionen*** und ***Variablen***). Zunächst müssen Sie – wenn Sie die Formel selbst eintippen wollen – im Reiter ***Formeln*** über die Schaltfläche mit dem Pluszeichen eine neue Formel erzeugen.



Textformeln geben einen Text zurück. In diesem Fall soll die Formel die passende Anrede zum Geschlecht des Patienten zurückgeben, so dass sich in allen Schriftstücken niemand mehr Gedanken darüber machen muss. Also startet die Formel schonmal mit dem in jedem Falle benötigten Textfragment „Sehr geehrte“, das zwischen Anführungszeichen gesetzt werden muss, um anzuzeigen, dass der Text keine Befehle enthält. Das Pluszeichen sorgt dafür, dass jede weitere Textausgabe einfach angehängt wird:

"Sehr geehrte" +

Innerhalb der Formel lassen sich ganz gewöhnliche Textvariable verwenden. Unsere Formel soll ja je nach Geschlecht des Patienten unterschiedliche Texte ausgeben. Also nutzen wir die Textvariable «**PatMW**», die je nach Geschlecht des Patienten die Buchstaben ***m***, ***w***, ***d***, ***x*** oder das Leerzeichen als Ergebnis zurückgibt. Genau das wird auch beschrieben, wenn Sie im Reiter ***Textvariablen*** diese Funktion anklicken.



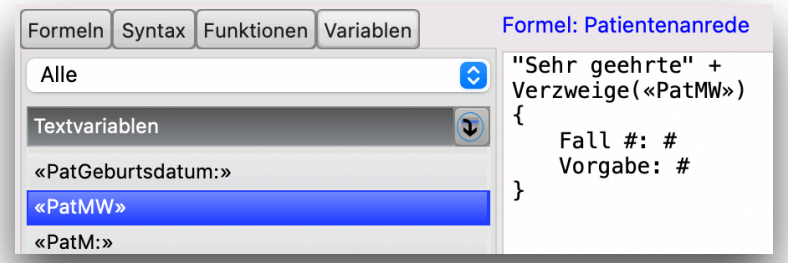
Zunächst gilt es aber, den passenden Befehl auszuwählen, der je nach dem Wert, den die «**PatMW**»-Variable zurückgibt, etwas anderes ausführt. Theoretisch lässt sich dies mit einer Verkettung von „Wenn dann“-Abfragen erledigen (Wenn das Geschlecht weiblich ist, tue dies, sonst: Wenn das Geschlecht männlich ist, tue das, sonst …). Eine „Wenn dann“-Frage gibt es natürlich, aber wenn es wie hier fünf verschiedene Fälle zu berücksichtigen gibt, ist dieses Verfahren zu umständlich. Der Befehl Verzweige (oder auch Switch) ist dafür viel praktischer. Schauen Sie im Reiter Syntax nach dem Befehl Switch und führen Sie einen Doppelklick darauf aus, um ihn in den Eingabebereich zu übernehmen.

"Sehr geehrte" +Switch(#) { Case #: # Default: # }

Zunächst einmal sollten Sie die Darstellung durch Hinzufügen von neuen Zeilen und Tabulator-Abständen etwas klarer machen, denn beides hat für DENTPORT keinerlei Bedeutung – es wird einfach überlesen. Und in der Beschreibung zum Switch-Befehl sehen Sie auch, dass es gleichwertige, deutsche Übersetzungen für die Befehlsworte gibt. Wenn Ihnen diese deutschen Versionen verständlicher erscheinen als die englischen Begriffe, können Sie diese nutzen:

```
"Sehr geehrte" +
Verzweige(#)
{
        Fall #: #
        Vorgabe: #
}
```

Überall dort, wo Sie jetzt noch ein Rautensymbol („**#**“) sehen, müssen Sie später etwas anderes einsetzen, also eine Abfrage oder Befehle. Die erste Raute, also die in den Klammern hinter dem Wort Verzweige, muss durch einen Wert ersetzt werden, der den Sprung zu verschiedenen Befehlen auslöst. In diesem Fall soll dafür ja der Inhalt der «**PatMW**»-Textvariable dienen. Also markieren Sie mit der Maus die erste Raute, gehen Sie in den Reiter mit den Textvariablen und führen Sie einen Doppelklick auf «**PatMW**» aus, um die Raute durch die Variable zu ersetzen.



Nun müssen wir natürlich definieren, wann was passieren soll. «**PatMW**» gibt Buchstaben zurück, nach denen wir nun unterscheiden müssen. Auch hier müssen eigentlich nur die beiden Rauten durch eine sinnvolle Formulierung ersetzt werden.

Fall "w": " Frau " + «PatNameGesamt»;

Ersetzen Sie einfach die erste Raute durch ein „w“ für das weibliche Geschlecht. Nach dem Doppelpunkt definieren Sie, was passieren soll bzw. was die Formel zurückgeben soll. Für das weibliche Geschlecht hängen Sie einfach " Frau " und den Namen der Patientin an, so dass insgesamt z.B. „Sehr geehrte Frau Erika Mustermann“ zurückgegeben wird – die Leerzeichen sind hier sehr wichtig. Das Semikolon schließt die Zeile ab und sorgt dafür, dass die Switch-Abfrage an dieser Stelle beendet wird, wenn der Fall „w“ zutrifft. Kopieren Sie nun diese Zeile und fügen sie zwei Mal an, um nun auch für männliche und unbestimmte Geschlechter die passenden Formulierungen zu hinterlegen.

Fall "m": "r Herr " + «PatNameGesamt»;

Fall "x": "(r) " + «PatNameGesamt»;

Damit sind drei von fünf Fällen für das Geschlecht definiert, zwei (das Leerzeichen und „d“ für divers) fehlen. Für alle verbleibenden Fälle nutzen wir die „Vorgabe“, die immer dann zutrifft, wenn kein anderer Fall eintritt.

Vorgabe: " Damen und Herren“;

In allen anderen Fällen gibt die Formel also die Formulierung „Sehr geehrte Damen und Herren“ zurück. Die komplette Formel sieht dann so aus:

```
"Sehr geehrte" +
Verzweige(«PatMW»)
{
        Fall "w": " Frau " + «PatNameGesamt»;
        Fall "m": "r Herr " + «PatNameGesamt»;
        Fall "x": "(r) " + «PatNameGesamt»;
        Vorgabe: " Damen und Herren";
}
```

Sie werden schnell bemerken, dass die Ausgabe dieser Formel auch nicht unbedingt richtig sein muss, denn eigentlich sollte gerade auch für diverse Geschlechter die oben nur für unbestimmte Geschlechter genutzte Formulierung zum Einsatz kommen. Das berühmte „Sehr geehrte Damen und Herren“ hingegen ist zum einen erst recht nicht politisch korrekt, zum anderen ist es als Anrede für Patienten (und nur dann enthält auch PatMW sinnvolle Werte) untauglich. Folgende Version der Formel ist also schon korrekter:

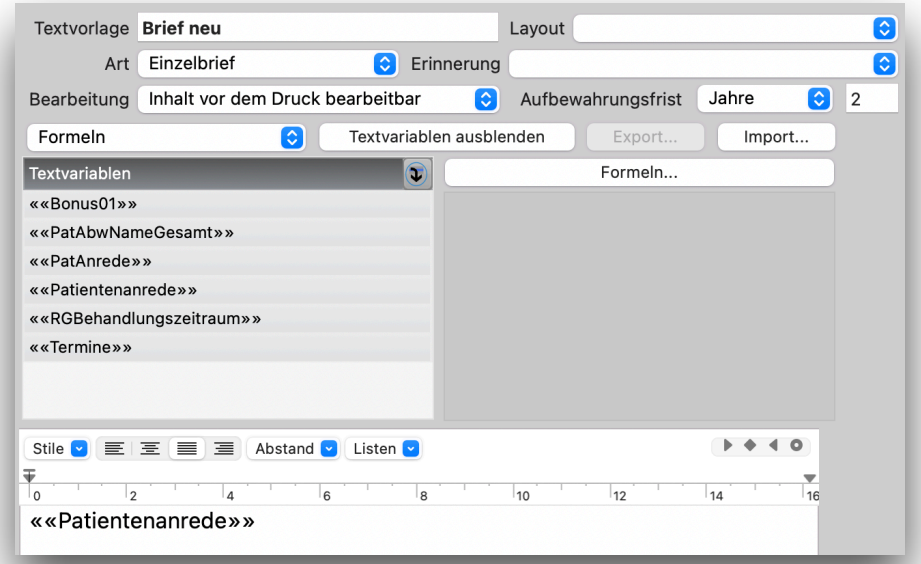
```
"Sehr geehrte" +
Verzweige(«PatMW»)
{
        Fall "w": " Frau " + «PatNameGesamt»;
        Fall "m": "r Herr " + «PatNameGesamt»;
        Vorgabe: "(r) " + «PatNameGesamt»;
}
```

Wenn Sie annehmen, dass die Formel fertig ist, klicken Sie auf die Schaltfläche „Formel testen“. DENTPORT schaut nun nach, ob die Formel sprachlich korrekt ist – in diesem Fall passiert einfach gar nichts und der Name der Formel wird in Blau dargestellt. Wenn DENTPORT die Formel nicht interpretieren kann, bekommen Sie eine Fehlermeldung, die eine Begründung enthält und an der Stelle, die DENTPORT nicht mehr verstehen kann, finden Sie die Inhalte in Rot. Die Fehlermitteilungen können Ihnen helfen, das Problem zu finden, führen aber oft auch in die Irre, weil sie sich auf Folgefehler beziehen können. Korrigieren Sie alle Fehler (im Beispiel haben wir das Pluszeichen vergessen) und

Erwartet: Case Default

```
"Sehr geehrte" +
Verzweige(«PatMW»)
{
    Fall "w": " Frau " + «PatNameGesamt»;
    Fall "m": "r Herr " «PatNameGesamt»;
    Vorgabe: "(r) " + «PatNameGesamt»;
}
```

testen Sie die Formel. Erst, wenn Sie auf „Formel übernehmen“ klicken, werden Ihre Änderungen auch für alle Anwender übernommen – es ist ja möglich, dass in der Praxis noch weitere Mitarbeiter die Formel nutzen. Und ein Klick auf Sichern sorgt schließlich dafür, dass Sie den Formeldialog verlassen.



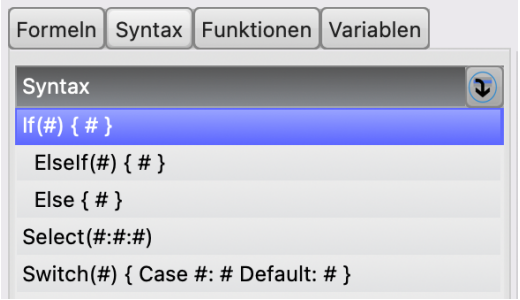
Bliebe noch, die Formel auch zu nutzen. Textformeln geben ja Texte zurück und sind so gesehen mit Textvariablen vergleichbar. Sie können die Formel überall dort nutzen, wo Sie auch Textvariable nutzen können. Erzeugen oder bearbeiten Sie also eine Textvorlage und wählen Sie im Aufklappmenü für die Textvariablen „Formeln“ aus. Dort sehen Sie die alle Textformeln und können diese per Doppelklick einsetzen. Im Unterschied zu normalen Textvariablen wird der lediglich Name der Formel in doppelte Anführungszeichen (Chevrons) gesetzt.

Sie können die oben entwickelte Formel auf unserer Website herunterladen und frei nutzen: https://dentport.de/Downloads/Formeln/Formel%20Patientenanrede.formel.zip.

Sprache bzw. Syntax

Die Sprache, mit der Formeln in DENTPORT angelegt werden, ist am ehesten mit einer sehr einfachen Skriptsprache vergleichbar – der Begriff des Programmierens ist dafür eher übertrieben. Sämtliche Syntax-Elemente stehen sowohl – wie beim Programmieren üblich – in englischer Sprache als auch auf Deutsch zur Verfügung. Wer schon einmal ein wenig programmiert hat, wird mit den englischen Varianten deutlich einfacher zurechtkommen, zumal sich die Namen so weit wie möglich an vorhandene Sprachen anlehnen. Wer hingegen mit dem Englischen Probleme hat, sollte die deutsche Version besser verstehen können. Alle Sprachelemente und Funktionen werden ausführlich und oft auch mit Beispielen erklärt. Verschaffen Sie sich einfach erstmal einen Überblick über die Möglichkeiten.

Die Grundzüge der Syntax sind sehr einfach und Sie sehen alle Bestandteile inklusive Erklärungen im Reiter „Syntax“. Wir beschränken uns im Folgenden auf die grundlegenden und am häufigsten verwendeten Elemente:



Grundlegendes

Immer dann, wenn mehrere Befehle nacheinander einen zusammenhängenden Block bilden sollen, umgeben Sie diesen mit geschweiften Klammern („**{**„ und „**}**“). Ihre Formel wird deutlich einfacher lesbar, wenn Sie die Zeilen zwischen den Klammern mit der Tabulatortaste etwas einrücken. DENTPORT überliest Tabulatoren, zusätzliche Leerzeichen und neue Zeilen übrigens, so dass Sie diese Zeichen nutzen können, um die Formel für sich verständlicher zu machen.

Einzelne Befehle schließen Sie mit einem Semikolon („**;**“) ab. Kommentare, die Ihnen später das Verständnis der Formel erleichtern, können Sie zwischen die Zeichenfolgen `/*` und `*/` setzen oder aber den doppelten Schrägstrich („`//`„) nutzen, der DENTPORT sagt, dass der komplette Rest der Zeile ignoriert werden soll.

Sie dürfen Innerhalb von Formeln auch Textvariable nutzen, für die es einen eigenen Reiter gibt und deren Namen zwischen Chevrons („«»“) gesetzt wird. Sie können auch vorhandene Formeln nutzen, deren Namen zwischen doppelte Chevrons („««»»“) gehört. Und Sie können auch eigene Variable definieren und nutzen, deren Name dann immer mit einem Dollarzeichen („**$**“) beginnt. Ganz wichtig: Um einer Variablen einen Wert zuzuweisen, nutzen Sie „**:=**", zum Vergleich hingegen „**==**".

Operatoren

Natürlich können Sie innerhalb von Formeln auch mit den vier Grundrechenarten („**+**“, „**-**", „*****“ und „**/**") rechnen. Texte hängen Sie mit dem Pluszeichen einfach aneinander.

Es stehen aber auch logische Operatoren zur Verfügung: „**==**" testet auf Gleichheit, „**!=**" dagegen auf Ungleichheit und es gibt natürlich auch Operatoren für kleiner, größer und dergleichen. Und es gibt logische Operationen wie „**and**“, „**or**“ oder „**xor**“ (nur eine der so verknüpften Aussagen ist wahr, aber nicht beide).

Abfragen

Die einfachste Bedingung für den Ablauf der Formel ist die einfache Wenn-Dann-Abfrage (**If(#) {#}**), die genau so angewendet wird, wie man es auch umgangssprachlich tun würde. Mit dem Sonst-Zusatz (**Else {#}**) oder auch der Kombination mit einer weiteren Bedingung (**ElseIf (#) {#}**) lassen sich weitere Fälle abbilden. Gewissermaßen die platzsparende Kurzform der Wenn-Dann-Abfrage ist die Auswahl (**Select(#:#:#)**).

Sollen mehr Fälle als nur wahr und falsch abgebildet werden, ist die Verzweigung (**Switch(#) {Case #:#}**) die bessere und leichter verständliche Wahl.

Schleifen

Formeln in DENTPORT können auch Schleifen enthalten, die mehrfach ausgeführt werden. Der Wiederhole-Befehl (**For(#:#) {#}**) sorgt dafür, dass der Programmteil in den geschweiften Klammern mehrfach durchlaufen wird. Wie oft das passiert, legen Sie im vorderen Teil selbst fest.

Funktionen

Funktionen führen Berechnungen aus oder erledigen andere Aufgaben mit den Werten, die ihnen in Klammern übergeben werden. Für Formeln stehen diverse Funktionen zur Verfügung, die sich per Aufklappmenü auf Themengebiete eingrenzen lassen:

Mathematisch

Im mathematischen Bereich gibt es alle Arten von Rechenfunktionen wie Sinus (**Sin()**) oder Quadrat (**Sqr()**). So lassen sich alle Berechnungen durchführen, für die Grundrechenarten nicht reichen.

Zeichenkette

Zum Verändern und Analysieren von Texten finden sich hier diverse Funktionen, beispielsweise auch zur Bestimmung der Länge eines Texts (**TextLength()**) oder zum Einfügen von Text (**Insert()**).

Datum und Uhrzeit

In den Reitern für Datum und Uhrzeit finden sich diverse Funktionen zum Auswerten von und Rechnen mit Zeiten und Datumsangaben. Manche dieser Funktionen haben übrigens zwei Formen, die sich in der Zahl der in Klammern übergebenen Parameter unterscheiden. So macht **Time(Z1:Z2)** aus Stunden (Z1) und Minuten (Z2) eine Zeit, **Time(Z1:Z2:Z3)** nutzt zusätzlich die übergebenen Sekunden (Z3).

Konvertierung

Hier finden sich Funktionen, die einen Datentypen in einen anderen umwandeln können. Beispielsweise versucht **TextToDate()**, aus dem übergebenen Text ein Datum zu machen.

Datenbank

Kurz gesagt: Als DENTPORT-Anwender sollten Sie sich nicht mit der Datenbankstruktur beschäftigen müssen, in der das Programm seine Daten speichert. Für eigene Textformeln benötigen Sie diese Funktionen normalerweise auch nicht. Es gibt die Funktionen zur Abfrage der Datenbank, damit die dentport GmbH mit maßgeschneiderten Formeln auf spezielle, nicht für jeden Anwender relevante Fragen reagieren kann. Um selbst sinnvolle Datenbankabfragen zu machen, müssten wir Ihnen zudem die – sehr umfangreiche und komplexe – Struktur der Datenbanken offenlegen, in der DENTPORT alle Daten speichert.

Datencontainer

Datencontainer kommen eigentlich nur im Zusammenhang mit Datenbankabfragen vor – das Resultat einer Datenbankabfrage ist immer ein Container. Einen Datencontainer kann man sich am ehesten wie eine Tabelle mit Spalten und Zeilen vorstellen. Sinngemäß gilt für Datencontainer aber das gleiche wie für Datenbanken: DENTPORT bringt diesen Mechanismus mit für individuell von der dentport GmbH auf den Anwender zugeschnittenen Formeln, die nicht unbedingt für jeden interessant und daher nicht als Funktion ins Programm integriert sind.

System

Im Bereich System finden sich Formeln für das Lesen und Schreiben von Dateien sowie noch einiges mehr. Diese Befehle sind natürlich nicht für Textformeln gedacht, sondern für Anwenderformeln, die den Befehlsumfang von DENTPORT erweitern. Normalerweise sollte niemand die Funktionen in diesem Bereich benötigen und einige davon können bei unsachgemäßem Gebrauch regelrecht Schaden anrichten. Wer sich dennoch herantrauen möchte, findet Beschreibungen zu allen Funktionen innerhalb des DENTPORT-Programms.

Formular

In diesem Bereich finden sich Formeln für selbstgestaltete Formulare (zu finden unter ***Stammdaten>Vorlagen>Formulare*** oder unter ***Stammdaten>System>Extra Daten***), mit denen Sie beispielsweise Farben von Elementen im Formular ändern, Warnungen hinzufügen oder Adressdaten übernehmen können.

Variable

In diesem Bereich stehen Ihnen alle Textvariablen von DENTPORT zur Verfügung, die Sie mit dem Aufklappmenü auf einzelne Bereiche wie Patienten oder die Praxis eingrenzen lassen. Sie können die Variablen innerhalb von Formeln frei nutzen und sogar andere Formeln innerhalb von Formeln nutzen.

Formeln für Texte

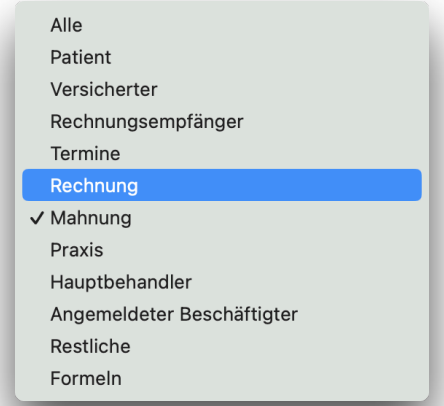
Unter den verschiedenen Arten von Formeln sind diejenigen, die Texte ausgeben und sich wie Textvariable einsetzen lassen, wohl am einfachsten zu verstehen und anzuwenden. Textformeln können Sie überall da definieren und nutzen, wo Sie auch Textvariable nutzen können. Mit Formeln können Sie auch komplexere Fragen stellen und unter Umständen einfacher zum Ziel kommen als mit den Variablen allein.

Textformeln müssen immer einen Text zurückgeben, entweder, indem Sie diesen direkt formulieren wie im Beispiel oben („*Sehr geehrte …*“) oder aber mit Hilfe des **Return**-Befehls. Es spricht für DENTPORT-Anwender einiges dafür, sich auf die Definition von Textformeln zu beschränken und alle weiteren Arten von Formeln außer Acht zu lassen, denn Anwenderformeln oder Formeln für Formulare setzen oft Wissen über die Datenstruktur von DENTPORT voraus.

Noch ein Beispiel für eine Textformel

Ein einfaches Beispiel hatten wir ja bereits zu Beginn, ein etwas anderes zeigen wir im Folgenden. Ziel der Formel ist es, den passenden Text für eine Rechnung auszugeben. Dabei geht es im Kern darum, ob die Behandlung in einem Termin erledigt werden konnte oder einen mehrtägigen Zeitraum gedauert hat.

Zunächst einmal suchen Sie einen der Bereiche auf, in denen Sie Textvorlagen für Rechnungen erstellen können (z.B. ***Stammdaten>Vorlagen>Rechnungen>Privat>Leistungsliste***) und wählen Sie dort einen Bereich, in dem Sie den erzeugten Text einfügen möchten (z.B. ***Fuß letzte Seite***). Klicken Sie auf ***Formeln***, erzeugen Sie mit dem Pluszeichen eine neue Formel und geben Sie ihr einen Namen – wir haben sie *RGBehandlungszeitraum* genannt. Im Vergleich zu den bisherigen Textformeln finden Sie im Reiter Variablen zwei neue Bereiche, nämlich ***Rechnungen*** und ***Mahnungen*** für Variable, die nur dann sinnvolle Werte liefern, wenn Sie gerade Rechnungen oder Mahnungen schreiben. Machen Sie sich zumindest grob mit den Möglichkeiten vertraut, die durch die entsprechenden Textvariablen entstehen.



Im Beispiel wollen wir den Text, den die Formel zurückgibt, in einer Variablen speichern. Daher sorgen wir erstmal dafür, dass der Inhalt dieser Variablen leer ist:

$rettxt := "";

Den Namen **rettxt** haben wir uns selbst ausgedacht, er sollte aber ausdrücken, was in der Variablen gespeichert werden soll. Das Dollarzeichen zu Beginn ist in DENTPORT Pflicht, um zu kennzeichnen, dass der Name einer Variablen folgt.

Nun prüfen wir erstmal, ob überhaupt ein Eintrag für eine abzurechnende Behandlung offen ist:

If (TextLength(«RgBehdlgVon:») > 0)

Wir starten also mit einer If-Abfrage und die Frage dazu wird in den Klammern dahinter gestellt. Wir nutzen die Funktion **TextLength()**, die als Argument die Textvariable «**RgBehdlgVon:**» erhält. Wenn der Beginn der Behandlung, also RgBehdlgVon einen Eintrag enthält, ist dessen Textlänge auch größer als Null und es werden die Befehle ausgeführt, die in geschweiften Klammern dahinter aufgeführt sind.

Innerhalb der If-Abfrage müssen wir nun herausfinden, ob die Leistung am gleichen Tag oder über einen längeren Zeitraum stattgefunden hat. Hierzu nutzen wir wiederum eine If-Abfrage, die nachsieht, ob Beginn und Ende der Behandlung am gleichen Tag stattgefunden haben:

```
If (TextToDate(«RgBehdlgVon:») == TextToDate(«RgBehdlgBis:»))
```

Wir nutzen jeweils die Funktion **TextToDate()**, um die in den Variablen **RgBehdlgVon** und **RgBehdlgBis** gespeicherten Texte in ein Datum zu verwandeln. Dieses Verfahren ist etwas sicherer als der direkte Vergleich der gespeicherten Texte miteinander. Der **==**-Operator sorgt für einen logischen Vergleich der Werte miteinander. Sind Beginn und Ende der Behandlung gleich, erzeugt die Formel folgenden Text:

```
$rettxt := "am " + «RgBehdlgVon:»;
```

Im Else-Zweig, also wenn die Beginn und Ende nicht identisch sind, ist der Rückgabetext natürlich ein anderer:

```
$rettxt := "vom " + «RgBehdlgVon:» + " bis " + «RgBehdlgBis:»;
```

Nach diesem äußeren Befehlsblock, also nach der schließenden Klammer, sorgen wir noch dafür, dass die Formel einen Wert zurück gibt. Das Zurückgeben eines Wertes ist für Textformeln Pflicht und erfolgt in diesem Fall über den **Return**-Befehl:

```
return $rettxt;
```

Im ersten Beispiel oben hatten wir einfach einen Text (*„Sehr geehrte“*) geschrieben und immer wieder erweitert, hier nutzen wir stattdessen den **Return**-Befehl. Dieser beendet die Formel unmittelbar – selbst wenn noch Befehle darauf folgen würden, würden sie nicht mehr ausgeführt.

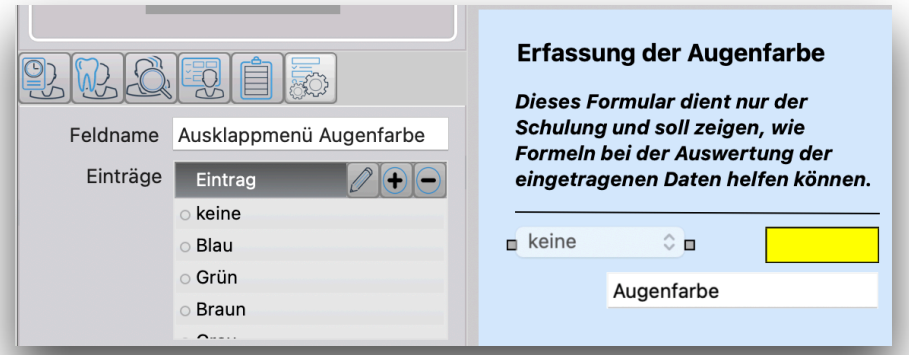
Die komplette Formel sieht am Ende also so aus:

```
$rettxt := "";
If (TextLength(«RgBehdlgVon:») > 0)
{
	If (TextToDate(«RgBehdlgVon:») == TextToDate(«RgBehdlgBis:»))
	{
		$rettxt := "am " + «RgBehdlgVon:»;
	}
	Else
	{
		$rettxt := "vom " + «RgBehdlgVon:» + " bis " + «RgBehdlgBis:»;
	}
}
return $rettxt;
```

Wie im ersten Beispiel testen Sie die Formel nun auf Fehler und sichern Sie sie, um sie später übernehmen zu können.

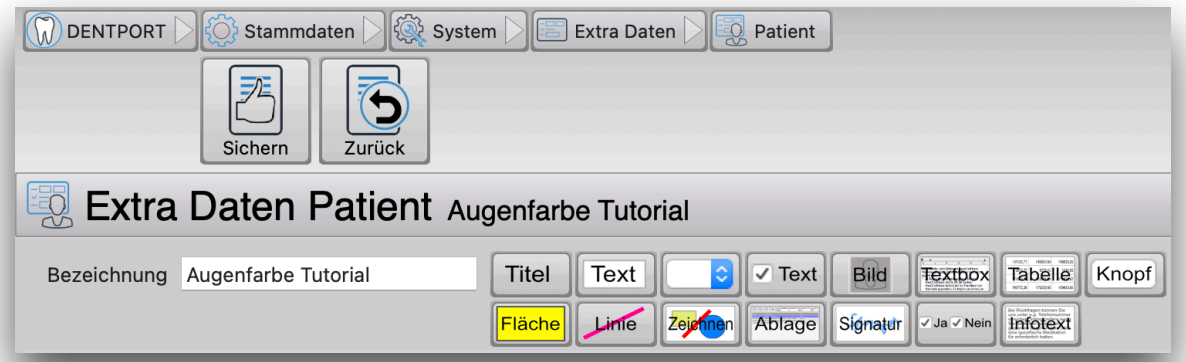
Sie können die oben entwickelte Formel auf unserer Website herunterladen und frei nutzen: https://dentport.de/Downloads/Formeln/RGBehandlungszeitraum.formel.zip.

Formeln für Formulare



Eine eigene Art von Formeln kann mit Daten arbeiten, die in Formularen (***Stammdaten>Vorlagen>Formulare***) angegeben oder in Form von Extra Daten (***Stammdaten>System>Extra Daten***) für jeden Patienten erfasst werden. Diese Art von Formeln hat einige Besonderheiten: Zum einen besteht innerhalb der Formel immer Zugriff auf die einzelnen Felder des Formulars, die unter den Variablen angezeigt werden. So lassen sich Eingaben überprüfen und Felder verändern. Zum anderen müssen Formeln für Formulare immer einen Wert zurückgeben, der sich aus dem Kontext ergibt. Ist die Formel mit einem Textfeld verbunden, wird erwartet, dass ein Text zur Beschriftung des Feldes zurückgegeben wird, Farbfelder erwarten hingegen eine Farbe und so weiter.

Wir zeigen die grundsätzlichen Möglichkeiten an einem Beispiel, das – zugegeben – etwas sinnlos ist, aber dafür sehr gut die Möglichkeiten zeigt. Wir erfassen mit einem Formular unter den Extra Daten die Augenfarbe des Patienten und zeigen diese danach in einem Farbfeld und in einem Textfeld an. Dafür müssen Formeln die Eingaben auswerten und die Werte der Felder entsprechend setzen.

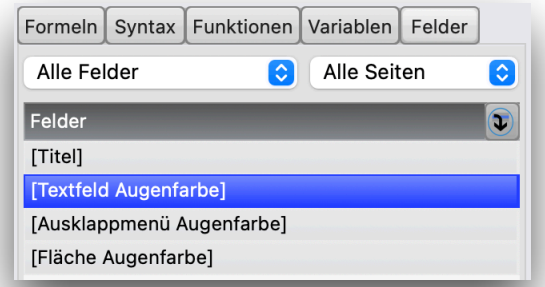


Legen Sie zunächst unter ***Stammdaten>System>Extra Daten>Patient*** ein neues Formular für Extra-Daten an, das wir im Beispiel „Augenfarbe Tutorial“ genannt haben und dass Sie später natürlich wieder werden löschen wollen. Ziehen Sie nun ein Aufklappmenü, ein Textfeld und eine Fläche in das Formular. Wichtig ist, den Feldern, die sie verändern oder abfragen wollen, einen sprechenden Namen zu geben, unter dem Sie sie später wiederfinden. Nennen Sie das Aufklappmenü also *„Aufklappmenü Augenfarbe“*, das Textfeld *„Textfeld Augenfarbe“* und die Farbfläche *„Fläche*

Augenfarbe“. Erzeugen Sie für das Aufklappmenü fünf neue Einträge namens „*keine*“, „*Blau*“, „*Grün*“, „*Braun*“ und „*Grau*“.

Für unser Beispiel wollen wir gleich mehrere Dinge mit Formeln erreichen: Eine Farbfläche hinter dem Aufklappmenü soll die ausgewählte Augenfarbe zeigen, wofür die erste Formel sorgt. Eine zweite Formel kümmert sich dann um das Textfeld, das zum einen die ausgewählte Farbe als Beschriftung zeigen soll und zum anderen seine Hintergrundfarbe entsprechend setzen soll.

Beginnen wir mit der Formel für die Farbfläche. Nach einem Klick auf „Formeln“ erzeugen Sie mit dem Pluszeichen eine neue Formel und vergeben Sie einen Namen dafür. Der erste Schritt besteht darin, sich die ausgewählte Zeile des Aufklappmenüs zu merken. Dieser Schritt ist in dieser Form nur beim Aufklappmenü nötig, das Programm-intern etwas speziell behandelt wird. Wir wollen sicherstellen, dass wir im Folgenden den Textinhalt des Aufklappmenüs nutzen können, also weisen wir diesen einer Variablen namens „Auswahl“ zu. Um sicherzustellen, dass diese wirklich einen Text enthält, weisen wir Auswahl einen leeren Text plus den Inhalt des Aufklappmenüs zu. Den Namen des Feldes, der in eckigen Klammern steht, müssen Sie übrigens nicht selbst tippen, sondern Sie können ihn auch im Reiter „Variablen“ unter „Felder“ finden und per Doppelklick in Ihre Formel einsetzen.



$Auswahl := "" + [Aufklappmenü Augenfarbe];

Eine zweite Variable ist für die Farbe zuständig, die wir an die Farbfläche zurückgeben wollen. Farben werden in DENTPORT als Zahl definiert. Wir weisen der Zahl über die Funktion ***ColorToInt(#:#:#)*** die Farbe Schwarz zu (0:0:0).

$Farbe := ColorToInt(0:0:0);

Nun weisen wir der Variablen „Farbe“ einen anderen Wert zu, der vom ausgewählten Eintrag im Aufklappmenü abhängt. Dafür nutzen wir wieder die Funktion Switch(), die alle Auswahlmöglichkeiten abhandelt und passende Farben zuweist. ColorToInt erwartet für die drei Grundfarben Rot, Grün und Blau jeweils eine Zahl zwischen 0 und 255. Die im Beispiel gewählten Werte sind natürlich etwas „bunt“, erfüllen aber ihren Zweck.

```
Switch($Auswahl)
{
        Case "Blau": $Farbe := ColorToInt(0:0:255);
        Case "Grün": $Farbe := ColorToInt(0:255:0);
        Case "Braun": $Farbe := ColorToInt(128:64:0);
        Case "Grau": $Farbe := ColorToInt(128:128:128);
}
```

Vielleicht ist Ihnen aufgefallen, dass es keine „Default“-Zeile gibt. Sie wird nicht benötigt, weil der Farbe ja in der ersten Zeile bereits Schwarz zugewiesen wurde. Trifft keine der „Case“-Zeilen zu, bleibt es einfach bei Schwarz. Anschließend muss die Farbe noch zurückgegeben werden:

```
return $Farbe;
```

Die komplette Formel sieht dann so aus:

```
$Auswahl := "" + [Aufklappmenü Augenfarbe];
$Farbe := ColorToInt(0:0:0);
Switch($Auswahl)
{
        Case "Blau": $Farbe := ColorToInt(0:0:255);
        Case "Grün": $Farbe := ColorToInt(0:255:0);
        Case "Braun": $Farbe := ColorToInt(128:64:0);
        Case "Grau": $Farbe := ColorToInt(128:128:128);
}
return $Farbe;
```

Nun muss diese Formel der Farbfläche zugewiesen werden. Dazu verlassen Sie den Formel-Editor und aktivieren Sie die Farbfläche. Im unteren Bereich der Optionen zur Farbfläche können sie der Farbfläche eine Formel zuweisen. Klicken Sie auf das kleine Pfeil-Symbol neben dem Namen der Formel, wählen Sie im Aufklappmenü „Formeln“ und dort die gerade erstellte Formel aus. Nun müssen Sie noch wählen, ob die Formel „einmalig“ oder „immer“ angewendet werden soll. Einmalig würde bedeuten, dass die Formel nach einer Änderung des Feldes angewendet wird. In diesem Fall allerdings ist „Immer“ die richtige Einstellung, denn das Feld, dem die Formel zugewiesen wird, ist ja kein Eingabefeld und kann deswegen nicht geändert werden. Die Einstellung „Immer“ sorgt dafür, dass nach jeder Änderung in einem beliebigen Feld des Formulars die Formel aufgerufen wird.

Zeit, die Formel zu überprüfen. Rufen Sie einen Patienten auf, klicken Sie auf „Extra Daten“ und dann auf das neu erzeugte Formular. Wenn Sie nun eine Augenfarbe auswählen, sollte diese sofort auch im Farbfeld übernommen werden. Falls es nicht funktionieren sollte, prüfen Sie zunächst den Inhalt der Formel und alle Einstellungen sowie die Namen der Felder, auf die sich ihre Formel bezieht. Wenn Sie Änderungen vornehmen, sollten Sie den Patienten ganz schließen, etwas anderes innerhalb von DENTPORT aufrufen und dann alles wieder neu laden, um sicherzustellen, dass auch die aktualisierte Formel genutzt wird.

Auf eine ganz ähnliche Weise legen Sie nun eine zweite Formel für das Textfeld an. In diesem soll entweder „Keine Angabe“ auftauchen, wenn keine Augenfarbe ausgewählt wurde oder aber eine Formulierung wie „Blaue Augen“. Eine zweite Formel wird benötigt, weil das Textfeld als Rückgabewert einen Text und keine Farbe erwartet. Hier die komplette Formel:

```
$Farbe := "" + [Aufklappmenü Augenfarbe];
If ($Farbe == "keine")
{
        $RetText := "Keine Angabe";
}
Else
{
        $RetText := $Farbe + "e Augen";
}
return $RetText;
```

In diesem Fall müssen nur zwei Fälle unterschieden werden, daher reicht hier die If-Abfrage. Wie zuvor wird diese Formel mit dem Textfeld verknüpft und die Ausführung wird auf „Immer“ gestellt, so dass die Formel nach jeder beliebigen Änderung aufgerufen wird.

Schließlich wollen wir noch zeigen, wie Formeln die Eingaben überprüfen können, die in Formularen gemacht werden. Wir werden eine Formel erstellen, die sicherstellt, dass das Formular nur dann gesichert werden kann, wenn eine Augenfarbe eingegeben wurde. Wenn nicht, erscheint eine Fehlermeldung und das Feld, in der eine Eingabe fehlt, wird markiert.

Dafür benötigen wir die Funktion **SaveOrPrintOK()**, die Sie im Bereich der Funktionen für Formulare finden. Diese Funktion darf nur in Formeln eingesetzt werden, die mit dem kompletten Formular verknüpft werden und der Bereich im darauf folgenden Block wird nur ausgeführt, wenn das Formular gesichert oder gedruckt werden soll.

```
If (SaveOrPrintOK())
{
        If ([Textfeld Augenfarbe] == "Keine Angabe")
        {
                WrongInputInField([Ausklappmenü Augenfarbe]:"Bitte geben Sie
                eine Augenfarbe an");
                return 0;
        }
}
return 1;
```

Innerhalb der Funktion müssen wir also prüfen, ob die Augenfarbe korrekt eingegeben wurde. Das kann auch indirekt passieren, indem wir den Inhalt des Textfeldes für die Augenfarbe daraufhin prüfen, ob dort noch „Keine Angabe“ steht. Wenn ja, nutzen wir die Funktion **WrongInputInField(#:#)**, um dem Anwender anzuzeigen, in welchem Feld eine Eingabe erwartet wird und warum das Formular so nicht gesichert werden kann. Dadurch, dass **return 0** folgt, wird das Sichern oder Drucken verhindert und das Formular bleibt geöffnet. Nur, wenn wie am Ende der Formel **return 1** genutzt wird, ist das Sichern des Formulars erfolgreich.

Sie können die Extra Daten „Augenfarbe“ auf unserer Website herunterladen und frei nutzen: https://dentport.de/Downloads/Vorlagen/Augenfarbe%20Tutorial.datenlayout.zip.

Anwender- und Hintergrundformeln

Während die bis hierher genannten Arten von Formeln primär dafür gedacht sind, dass Sie als Anwender des Programms dessen Möglichkeiten durch Eigeninitiative erweitern können, verhält es sich mit den Anwender- und Hintergrundformeln, die wir Ihnen im Folgenden vorstellen möchten, anders. Sie dienen dazu, den Funktionsumfang des DENTPORT-Programms zu erweitern und neue Funktionen zu schaffen, die nicht für alle Anwender, sondern oft nur für einen Bruchteil davon interessant sind. Da Anwender- und Hintergrundformeln fast immer auch Zugriff zur Datenbankstruktur des Programms benötigen, kommt auch fast ausschließlich die dentport GmbH selbst als Entwickler solcher Formeln in Frage.

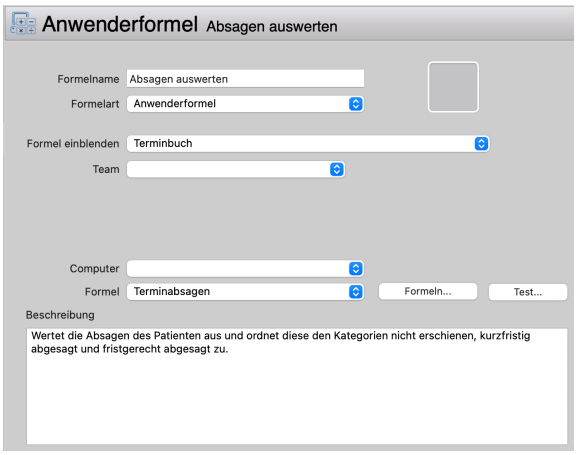
Kurz gesagt: Anwender- und Hintergrundformeln sind normalerweise nichts, was Sie als Anwender entwickeln könnten, vielmehr beantworten wir mit diesen bisweilen Fragen nach Features, von denen wir wissen, dass nur wenige diese nutzen würden. In diesen Fällen wäre es nicht sinnvoll, die Funktion fest ins Programm zu integrieren und damit bei vielen DENTPORT-Anwendern eher Verwirrung zu stiften als ihnen einen Mehrwert zu bieten. Derartige Formeln bieten wir auch zum Herunterladen an, wobei Sie als Kunde sich genau ansehen sollten, was die Formel tut und ob Sie diese benötigen. Sinn dieses Kapitels ist es damit vor allem, für Sie verständlicher zu machen, wie diese Formeln arbeiten und Ihnen zu zeigen, wie Sie womöglich kleine Anpassungen vornehmen können. Auch ein Wort der Warnung sei erlaubt: Formeln können natürlich auch Daten verändern, was unerwartete und unangenehme Folgen haben kann.

Anwender- und Hintergrundformeln werden unter ***Stammdaten>System>Praxis>Formeln*** definiert. Der grundsätzliche Unterschied ist, dass Anwenderformeln eine eigene Schaltfläche bekommen und somit nur dann aktiv werden, wenn diese Schaltfläche auch betätigt wird. Hintergrundformeln hingegen können zu bestimmten Zeiten und Gelegenheiten laufen und im Hintergrund Daten überprüfen und auch manipulieren – vor allem letzteres bringt natürlich einige Möglichkeiten für potentielle Probleme mit sich.

Ein Beispiel für eine Anwenderformel

Wir sehen uns aber eine einfache Anwenderformel an, die tatsächlich von einem DENTPORT-Anwender gewünscht wurde. Der Anwender wollte die abgesagten Termine von Patienten auswerten können, um daraus eine Aussage über deren Zuverlässigkeit ableiten zu können. Diese Funktion ist

nur für wenige DENTPORT-Anwender interessant und wurde noch nicht häufig gewünscht, gleichzeitig lässt sich diese Möglichkeit perfekt über eine Formel realisieren.



Legen Sie unter ***Stammdaten>System>Praxis>Formeln*** eine neue Formel an und nennen diese beispielsweise „Absagen auswerten“. Als Typ wählen Sie ***Anwenderformel***. Wenn Sie in das entsprechende Feld eine eigene Grafik einsetzen, wird diese für die Schaltfläche genutzt, mit der Sie die Formel später aufrufen können. Im Aufklappmenü „Formel einblenden“ wählen Sie nun aus, an welcher Stelle im Programm die Schaltfläche für die neue Formel angezeigt werden soll. Hier sind so gut wie alle Bereiche des DENTPORT-Programms verfügbar. Da wir eine Formel zur Auswertung der Termine anlegen wollen, wählen wir „Terminbuch“ aus. Die Formel selbst besteht eigentlich aus zwei Teilen, wobei im ersten die Abfrage an die Datenbank von DENTPORT definiert wird:

```
DBNewSelection();
DBAddColumn("id":"termin");
DBAddColumn("status":"termin");
DBAddWhere("patientID":"termin":0:«PatNummer»);
DBAddWhere("patienttype":"termin":8:"(1, 2)");
DBAddWhere("terminmode":"termin":0:1);
DBAddWhere("status":"termin":8:"(4, 5, 10)");
DBAddWhere("dualterminID":"termin":0:0:"(":"");
DBAddWhere("dualmaster":"termin":0:1:"or":")");
$tcont = DBSelectContainer();
```

DBNewSelection() bereitet die Abfrage der Datenbank vor, **DBAddColumn()** fügt die Spalten *„id“* und *„status“* aus der Tabelle *„termin“* der Abfrage hinzu. Die vielen Aufrufe von **DBAddWhere()** definieren die eigentliche Suche nach abgesagten Terminen. Schließlich führt **DBSelectContainer()** die Abfrage aus und übergibt das Ergebnis in Form eines Datencontainers an eine Variable.

Diese Erklärung ist nur grob und kann auch nicht anders ausfallen, weil die Definition der Abfrage tiefe Kenntnisse der Datenbankstruktur von DENTPORT erfordert. Im Speziellen muss der Autor der Formel wissen, dass Termine in einer Datenbank-Tabelle namens *„termin“* gespeichert werden, welche Felder diese beinhaltet und wie man nach abgesagten Terminen sucht. Als DENTPORT-Anwender können und müssen Sie dies nicht wissen – auch innerhalb der dentport GmbH haben nicht alle einen Überblick über die Datenbank-Struktur und Formeln wie diese hier haben ihren Ursprung fast immer in der Entwicklungsabteilung. Der Rest der Formel ist dagegen deutlich einfacher zu verstehen:

```
$nicht = 0;
$kurz = 0;
$frist = 0;
$rows = ContainerRows($tcont);
For($i:$rows)
{
        Switch(TextToNumber(ContainerCell($tcont:"status":$i)))
        {
                Case 4: $nicht = $nicht + 1;
                Case 5: $kurz = $kurz + 1;
                Case 10: $frist = $frist + 1;
        n}
}
```

Zunächst werden Variable angelegt, die die gewünschten Terminabsagen enthalten. Dann wertet eine Schleife alle Zeilen der Tabelle (also des Datencontainers) aus und zählt dabei die entsprechenden Werte für Terminabsagen hoch.

```
$txt = "Abgesagte Termine Patient " + «PatNameGesamt» + ":\n" +
                "  Nicht erschienen: " + NumberToText($nicht) + "\n" +
                "  kurzfristig Abgesagt: " + NumberToText($kurz) + "\n" +
                "  fristgerecht Abgesagt: " + NumberToText($frist);
Question($txt:"OK");
return "";
```

Schließlich wird in einer Variablen namens $txt ein Text mit den Ergebnissen der Auswertung gespeichert, der von der Funktion **Question()** als Dialog dargestellt wird. Dabei bedeutet **\n** übrigens *Neue Zeile*.

Wie gesagt, ohne Kenntnisse der Datenbankstruktur und der Art wie DENTPORT darin Daten speichert (in diesem Fall abgesagte Termine) ist es vollkommen undenkbar, derartige Formeln zu erstellen. Es sollte Ihnen aber nun möglich sein, diese Formeln zumindest grob zu verstehen und eventuell zu modifizieren. Beispielsweise sollte es leicht möglich sein, die Formel nur nach nicht wahrgenommenen oder kurzfristig abgesagten Terminen suchen zu lassen, statt auch fristgerechte Absagen zu berücksichtigen (die übrigens den Wert „10“ als Status haben – kleiner Tipp, falls Sie es versuchen möchten).

Wir haben ganz bewusst ein Beispiel für eine Anwenderformel gewählt, die lediglich den vorhandenen Inhalt der Datenbank auswertet. Es ist aber rein technisch auch möglich, Werte in die Datenbank zurück zu schreiben. Diese

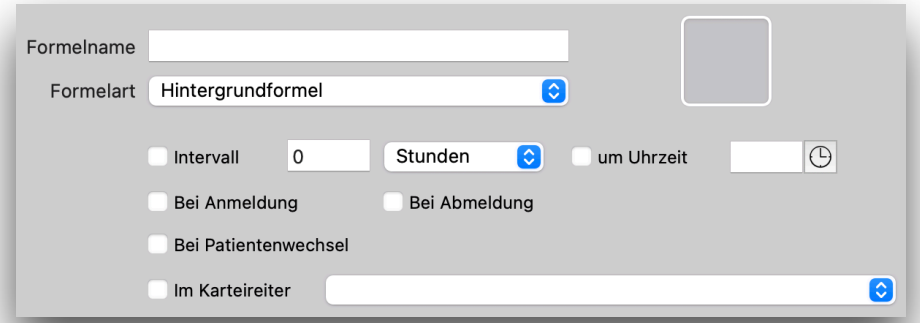
Sie können die Anwenderformel für Terminabsagen auf unserer Website herunterladen und frei nutzen: https://dentport.de/Downloads/Formeln/Terminabsagen.anwformel.zip.

Veränderungen können aber zu umplausiblen Inhalten und damit zu Fehlern führen, weshalb wir Ihnen sehr dringend nahelegen, keine eigenen Experimente in dieser Richtung zu machen.

Hintergrundformeln

Eine Unterart von Formeln ist dafür gedacht, im Hintergrund zu laufen, also während das Programm ganz normal für andere Aufgaben zur Verfügung steht. Auch Hintergrundformeln sind überwiegend für den Gebrauch durch die DENTPORT-Entwickler gedacht, die darüber Funktionalität zur Verfügung stellen können, die für manche Anwender wichtig, für die breite Masse aber ohne Belang ist. Auch hier gilt also: Die Erklärung ist vor allem dafür gedacht, Formeln aus externer Quelle zu verstehen und weniger dafür, eigene Formeln zu entwickeln.

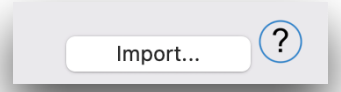
Hintergrundformeln können zu bestimmten Gelegenheiten oder in Zeitabständen, die Sie festlegen können, Funktionen ausführen. Eine typische Anwendung dafür besteht darin, die Eingaben durch Benutzer oder den Inhalt der Datenbank zu überprüfen, Fehler zu melden oder Korrekturen in die Datenbank zurück zu schreiben. Schon daran können Sie sehen, dass Hintergrundformeln nicht für experimentierfreudige Anwender, sondern eher für die Programm-Entwickler gedacht sind.



Allerdings ist es wie immer sinnvoll, sich anzusehen, was möglich ist. Und da sind vor allem die Optionen zu nennen, sobald Sie die Art in eine Hintergrundformel ändern. Über Ankreuzfelder, von denen Sie auch mehrere gleichzeitig auswählen können, bestimmen Sie, ob die Formel in regelmäßigen Intervallen, zu bestimmten Zeiten, bei An- oder Abmeldung, beim Wechsel des Patienten oder bei Aufruf eines bestimmten Karteireiters ausgeführt werden soll. Auf diese Weise sind beispielsweise Überprüfungen der Eingaben möglich, ohne dass die Formel unnötig oft ausgeführt werden muss.

Im- und Export von Formeln

Beim Im- und Export von Formeln muss man mehrere Fälle unterscheiden: Formeln für Formulare und Extra-Daten lassen sich nicht einzeln sichern oder öffnen, vielmehr gehören sie immer zum jeweiligen Formular oder den Extra-Daten, zumal sie sich ja auch auf dessen Inhalte beziehen. Um diese Formeln zu im- und exportieren muss also das jeweilige Formular beziehungsweise die Extra-Daten mitsamt deren Layout und allen Formeln ex- oder importiert werden. Alle darin verwendeten Formeln werden in der Datei gespeichert und stehen nach dem Import auch anderen zur Verfügung.



Für alle anderen Fälle, also Text-, Anwender- und Hintergrundformeln ist der Formeldialog selbst zuständig. Öffnen Sie zunächst den dazu gehörigen Formeldialog – ganz so, als wollten Sie eine neue Formel

anlegen. Für den Import von Formeln klicken Sie auf die ***Import***-Schaltfläche. Danach wählen Sie die Datei, die die Formel enthält und importieren Sie sie.

Für den Export von Formeln gibt es dagegen keine direkt zugängliche Schaltfläche. Stattdessen zeigt sich die Export-Möglichkeit nur dann, wenn Sie beim Klicken auf die ***Import***-Schaltfläche gleichzeitig die Alt-Taste (Windows) beziehungsweise Option-Taste (Mac) gedrückt halten. Dann erscheint ein Dialog, der fragt, ob man eine Formel im- oder exportieren möchte. Nur dann, wenn Sie hier ***Markierte Formel exportieren*** klicken, lässt sich die Formel auch wirklich in einer Datei sichern.

Dass der Export etwas versteckt ist, hat gute Gründe: Speziell Anwender- und Hintergrundformeln sind wie bereits gesagt nicht dafür gedacht, von den DENTPORT-Anwendern selbst entwickelt zu werden. Vielmehr ist diese Möglichkeit für Erweiterungen des DENTPORT-Programms gedacht, die nicht für alle Anwender interessant sind und werden eher von DENTPORT-Entwicklern als von den Anwendern genutzt. Schon deshalb sollten Sie übrigen beim Import von Formeln vorsichtig sein und die Quelle genau prüfen.

Bei Textformeln ist die Gefahr natürlich etwas geringer, denn sie enthalten nur eine andere Form von Textvariablen. Hier gibt es allerdings einen anderen Zusammenhang zu beachten: Enthalten Textvorlagen auch Textformeln, müssen diese natürlich ebenfalls gesichert und vor der Vorlage importiert werden, weil ansonsten nicht klar ist, auf welche Formel sich die Vorlage bezieht. Wenn Sie Textvorlagen weitergeben möchten, sollten Sie also auch alle darin enthaltenen Textformeln weitergeben.

Liste der Versionen:

- Version 1.0 vom 16.10.2024 – erste Version